

Talent



Creado por LOGO Computer Systems Inc., Canadá

Redacción del manual FERNANDEZ LONG Y REGGINI S.A.

MSX-Logo

Telemática S.A.

Versión de las palabras Logo al español
y redacción del Manual:
Fernández Long y Reggini S.A.
Av. Alem 1074 - Tel. 311-8413 y 312-7823
1001 Buenos Aires, Argentina

Chile 1347 - Tel. 37-0051 al 54
1098 Buenos Aires, Argentina

I.S.B.N. 950-9688-02-7

Manual desarrollado en la Argentina bajo licencia de:
LCSI
Logo Computer Systems Inc.
Montreal, Canadá H8T 1A1

Impreso en Argentina
Printed in Argentina

Se terminó de imprimir en el
mes de octubre de 1986 en los talleres gráficos
de Sebastián de Amorrortu e Hijos S.A.
Esteban de Luca 2241 - Buenos Aires - Argentina

MSX-Logo

LCSI LICENCIA ESTANDAR DE USUARIO FINAL

Licencia:

EL OTORGANTE concede al usuario final una licencia consistente en el derecho a utilizar el "software" de acuerdo con los siguientes términos:

1. El usuario final sólo podrá emplear el "software" en un solo sistema de computadora en cada momento dado.
2. Se permite al usuario final la transferencia a terceros de esta licencia y del "software" siempre que;
   (a) Dicho tercero se avenga a todos los términos del presente acuerdo.
   (b) El usuario final no retenga en su poder copia alguna del "software".

Licencia Telemática S.A. y LCSI de usuario final:

Telemática S.A. no asume ningún tipo de responsabilidad sobre el uso y aplicaciones de este software, o por errores en este manual o en su software. Puede existir, sin embargo, soporte adicional y/o garantías extras de parte de LCSI sobre este software. Telemática no tiene ninguna relación en ello, y no asume ninguna responsabilidad o garantía relacionada con lo antedicho.

Este manual está sujeto a cambios sin previo aviso y no constituye una obligación para Telemática S.A. informar sobre estos cambios.

PROLOGO

Esta obra introduce al lector a una nueva implementación de Logo.

MSX-Logo significa otro hito en el camino del desarrollo de versiones de Logo cada vez más poderosas. Incluye la posibilidad de dibujar en la totalidad de la pantalla, pintar o sombrear recintos, utilizar distintos colores simultáneamente, el control de tortugas móviles, la manipulación de palabras y listas, el empleo de listas de propiedades y la generación de música.

Pueden activarse treinta tortugas simultáneamente (antes llamadas "actores" en otras versiones de Logo), con distintas figuras y colores y cada una de ellas puede dejar rastro al moverse. Existen además las órdenes cuando y al. contacto, que desencadenan una serie de acciones si se produce un suceso determinado. Es como si un diablillo o "demonio" estuviera al acecho mientras se ejecuta un procedimiento cualquiera; cuando ve que se produce el hecho especificado, hace que se cumplan las órdenes consignadas.

La sigla MSX designa un conjunto de normas que hacen posible que un programa pueda operar en distintos modelos de computadoras, siempre que hayan adoptado las normas MSX citadas.

Por qué Logo

El aprendizaje del lenguaje Logo es una de las puertas más interesantes para acceder al mundo de la computación. Su sencillez, su potencia y la posibilidad de emplearlo en todos los campos de la actividad humana pueden llegar a convertirlo en el lenguaje de programación más difundido y aceptado en los próximos años.

La potencia de Logo se debe a tres razones fundamentales. En primer lugar, sus principios son fáciles de aprender y los que se inician logran de inmediato resultados interesantes, sin necesidad de preparación esmerada. Por ello, incluso los niños pueden usarlo y, de hecho, millares de ellos lo están haciendo actualmente. Es

por esta circunstancia que las computadoras, con Logo, se convierten en herramientas útiles para el proceso educativo. Sin embargo, Logo no es un lenguaje solo para niños. Esta imagen, que existe en algunos círculos, se debe al hecho de que los primeros ejemplos de su empleo ilustran procedimientos elementales para niños y/o principiantes adultos carentes de conocimientos previos de computación. Utilizarlo luego en aplicaciones importantes requiere, por supuesto, estudios organizados.

La segunda razón reside en su "extensibilidad", que hace de él un lenguaje descriptivo de extraordinaria utilidad. Asimismo, la formulación de programas en Logo no solo imparte una jerarquización lógica, sino que hace sencilla la búsqueda de los errores que inexorablemente aparecen en la programación.

La tercera razón tiene que ver con el uso de estructuras de datos muy generales como las listas, y las posibilidades que brinda la recursión.

Las nuevas versiones de Logo disponibles, como la presente, nos acercan hoy más a la posibilidad de su uso en campos diversos.

Las computadoras en la educación

El uso de Logo en la educación presupone una visión distinta de las máquinas. Logo se relaciona íntimamente —como modalidad educativa— con la creatividad y la libertad, valores que hay que reencontrar y reconstruir permanentemente en todo ámbito.

La implementación de Logo en las aulas no puede improvisarse, sino que requiere por parte de los docentes y de los directivos un reconocimiento cabal de los principios cognitivos en que está inspirado. Los educadores deben comprender que la filosofía de su uso es más importante que los productos que se pueden obtener de la máquina. Como instrumento delicado, Logo torna más rico y eficaz el proceso educativo pero, al mismo tiempo, lo vuelve más complicado y difícil de encaminar si no existe una concepción clara de sus fundamentos y alcances.

Es preciso comprender que un aprendizaje verdadero y profundo solo se alcanza cuando el conocimiento formal se relaciona con el conocimiento personal de manera variada y diversa. En un proceso educativo genuino, los temas tratados deben poseer puntos de contacto con el mundo circundante y relevancia personal para el alumno. El diálogo enseñanza-aprendizaje debe alentar y favorecer la actividad autoconstructiva del educando, a fin de que éste se encuentre con su propio yo, saque partido de sus recursos

intelectuales y perciba que lo que aprende lo habilita para interactuar con su alrededor. Deben existir lazos de afectividad con lo que se aprende; es necesario que lo conceptual se vincule claramente con lo instrumental y que todo ambiente de aprendizaje permita convertir las ideas en hechos, con la posibilidad de corregir errores y de modificar rumbos. Logo solo puede arraigar y crecer en un ámbito donde se aceptan y practican las premisas anteriores.

Logo puede ser considerado como un aporte filosófico a la educación * lo que agrega a sus posibilidades poderosas en el campo de la computación, una componente social trascendente. Es un camino del pensamiento que busca que la ciencia y la tecnología estén más al servicio de las personas y de la cultura.

HORACIO C. REGGINI

Buenos Aires, marzo de 1986

* Desafío a la mente, Seymour Papert, Buenos Aires, Galápago, 1981, Dist. Emecé.

INTRODUCCION

El manual MSX-LOGO

Este manual MSX-Logo se estructura en tres partes. Los primeros capítulos introducen rápidamente en las ideas del lenguaje Logo y en la manera de escribir procedimientos Logo. Para su presentación se ha utilizado una secuencia similar a la de otras publicaciones de introducción a Logo, comenzándose con el lenguaje de la tortuga.

Los capítulos siguientes describen los agentes o tortugas móviles, el manejo de las palabras y listas, la realización de diseños tridimensionales con la tortuga espacial y, finalmente, la creación de composiciones musicales.

*Esto no significa que sea la única secuencia posible para el aprendizaje; solo intenta dar un panorama de la estructura y las posibilidades de Logo. Quien desee profundizar, puede consultar el libro Alas para la Mente *, donde encontrará numerosos ejemplos.*

Los dos últimos capítulos contienen una lista completa de las primitivas y sus funciones, así como un listado de los mensajes de pantalla. Las primitivas están ordenadas alfabéticamente facilitando la rápida consulta del listado.

Finalmente, una serie de anexos contiene tablas, especificaciones acerca del uso del teclado y una lista de las primitivas agrupadas según su función. Esta sección del manual está pensada para servir de consulta permanente.

La versión MSX-Logo en castellano de la correspondiente versión MSX-Logo en inglés ha sido realizada eligiendo palabras de uso habitual y corriente en el mundo de habla hispana. En general, se han mantenido los vocablos ya establecidos en publicaciones y versiones Logo conocidas desde hace varios años y aceptadas por diversos círculos, con pequeños cambios que se señalan en el Anexo E.

*** Alas para la Mente, Horacio C. Reggini, Buenos Aires Galápago, 1982 Emecé.**

Es así como esta versión mantiene la forma infinitiva para los verbos, por las razones que se detallan más adelante. La condición de dibujo de la tortuga al moverse sigue llamándose metafóricamente conpluma (y su contraria, sinpluma) no obstante que nadie en la actualidad usa una "pluma" para escribir o dibujar. El uso de "lápiz, lapicera, bolígrafo, tiza, fibra", etc., tendría connotaciones más locales o temporales que el antiguo término "pluma" que es figurativo. El uso de los prefijos "con" y "sin", —que también se emplean para otras órdenes— es preferible a utilizar los vocablos "bajar" o "descender" y "subir" o "alzar". Además conpluma y sinpluma son términos más adecuados para el caso de la tortuga tridimensional que puede adoptar diferentes posiciones en el espacio.

*En las explicaciones, nos referimos a la tortuga (o a la computadora o a Logo) como si fuese un ser viviente: "le hablamos, la hacemos mover, nos responde"... Lo importante no es la tortuga o la computadora, sino lo que nosotros podemos lograr que ella haga y nuestra reflexión sobre la forma de lograrlo. Un estudio lúcido sobre la actitud de las diferentes personas frente a la computadora y sobre cómo ésta ha influido en la manera de pensarnos a nosotros mismos se puede ver en el libro El Segundo Yo *.*

Acerca del uso del infinitivo

El uso del modo infinitivo de los verbos —y no de otras formas personales— en esta versión castellana de MSX-Logo obedece a las siguientes razones:

- *El modo infinitivo es en castellano una de las formas de comunicar una indicación o una orden. Se dice así por ejemplo, "no girar a la izquierda", "agitar antes de usar" o "servir bien frío". Esta manera de usar el infinitivo en castellano no se emplea en la lengua inglesa.*
- *Los procedimientos Logo definidos por el usuario tienen a menudo formas verbales (por ejemplo, para caminar); describen una acción determinada —en este caso, los movimientos necesarios para realizar la acción de caminar—. De ahí también el uso del infinitivo. Las formas verbales de las órdenes primitivas (repetir, esperar, borrar, etc.), siguen un criterio similar y tienen en cuenta que es el usuario quien le ordena hacer algo a la computadora o a la tortuga.*
- *La forma impersonal del infinitivo permite acomodarse a una audiencia más amplia, ya que en castellano existe una diferenciación entre el trato familiar del "tú" —que sería el más apropiado para la audiencia infantil—, y el "usted" que se estila entre la gente adulta. Además, como forma impersonal, es común a todos los países de habla hispana, y permite dejar de lado los regionalismos de algunas formas personales (por ejemplo el uso del "vos" en los países del Río de la Plata).*
- *Por último, el infinitivo permite resolver los problemas que se plantean en algunos casos debido a los accidentes verbales de número y tiempo. Por ejemplo, al hablar a más de una entidad Logo (el caso de las tortugas múltiples) habría que usar la forma plural de los verbos si no se empleara el modo infinitivo.*

*** El Segundo Yo, Sherry Turkle, Buenos Aires, Galápago, 1984, Dist. Emecé.**

CAPÍTULO 1
LA COMPUTADORA TALENT-MSX

Este capítulo es una introducción a la instalación y uso de la computadora Talent-MSX. Comprende la explicación de su teclado y de cómo utilizar los periféricos. Indicaciones complementarias se dan también en los manuales de cada dispositivo.

El conjunto necesario para utilizar MSX-Logo está formado por el teclado (que es la computadora misma), un televisor común o un monitor, y el cartucho Logo. También será imprescindible disponer de un grabador común de casetes o una unidad de disco para poder almacenar de manera permanente los trabajos realizados.

Conexión de los dispositivos

Las conexiones entre los dispositivos deben realizarse con todos los equipos apagados.

En primer lugar, se debe conectar el cable de TV en el conector RF situado en la parte posterior de la computadora, por un lado, y en la antena del televisor, por el otro. El cable suministrado es de 75 ohms de impedancia; si la antena del televisor fuera de 300 ohms de impedancia, deberá agregarse un adaptador.
El cable del grabador tiene en uno de sus extremos una única ficha, que debe enchufarse en el conector ubicado en el costado derecho de la computadora. En el otro extremo, el cable presenta tres terminales que deben conectarse en el grabador de la siguiente manera:

conector blanco:	a la salida EARPHONE o MONITOR
conector rojo:	a la entrada MIC o EXT MIC
conector negro:	a la entrada REMOTE

Nota: En algunos grabadores la polaridad de la entrada REMOTE es distinta a la usada por la computadora. En este caso, se deberá intercalar un inversor de polaridad o desconectar ese cable; en este último caso no se producirá el arranque o detención automática del motor del grabador.

Si hay una unidad de disco, se la debe ubicar a 10 ó 30 cm de la computadora: su conexión se debe realizar en la ranura de expansión situada en la cara posterior de la computadora, junto al costado derecho; el conector extremo del cable

de la unidad de disco se debe introducir en la ranura en posición horizontal y con las muescas hacia arriba, de forma que ingrese en su totalidad. A continuación, se deben cerrar las palanquitas laterales que inmovilizan el cable. Hay que tener en cuenta que si el cable se mueve o desconecta, la computadora Talent no responderá a las instrucciones y será necesario apagarla, ajustar bien las conexiones y volver a encenderla, perdiéndose el contenido de la memoria de la computadora.

Conexión del cartucho del lenguaje Logo

El lenguaje Logo está grabado de manera permanente en un cartucho ROM que se conecta a la máquina. Mientras se encuentre instalado, el lenguaje Logo aparecerá automáticamente en la computadora cada vez que ésta se encienda.
El cartucho DEBE COLOCARSE ÚNICAMENTE CON LA COMPUTADORA APAGADA; de otra manera, se dañará su contenido.
El cartucho se debe introducir en la ranura ubicada en la parte superior de la consola, con su título hacia el frente y LA HENDIDURA DEL CARTUCHO DE CARA A LA PARTE POSTERIOR DE LA COMPUTADORA; nunca se lo debe introducir con la hendidura hacia adelante.
Para retirarlo ES NECESARIO APAGAR LA COMPUTADORA PREVIAMENTE.

Puesta en marcha

Como regla general, todos los dispositivos deben encenderse antes que la computadora; esto es fácil de controlar observando las luces indicadoras de encendido de que disponen casi todos los equipos.
Hay que sintonizar el televisor en los canales 2, 3, ó 4. Si la sintonía es correcta, y se encuentra conectado el cartucho Logo, aparecerá el siguiente mensaje en la pantalla:

```
MSX system
Version 1.0
Copyright 1983 by Microsoft
```

y a continuación:

```
(c) Copyright Logo Computer
Systems Inc. 1985
Bienvenido a MSX-Logo
?
```

Cómo preparar un disco

El disco es el elemento más adecuado para guardar y recordar procedimientos, variables y archivos.
Antes de poder utilizar un disco por primera vez, es necesario prepararlo para que sea aceptado por el sistema operativo MSX-DOS (del inglés, Disk Operating System, Sistema Operativo de Discos). En la jerga de la computación, a esta operación se la llama "inicializar" o "formatear" un disco y es realizada por el comando **format**, que es una orden del sistema operativo MSX-DOS.

Para trabajar con el sistema operativo se debe retirar el cartucho Logo (SIEMPRE CON LA COMPUTADORA APAGADA). Luego, se debe introducir el disco del sistema operativo en la unidad de disco, encenderla y, a continuación, encender la computadora. En la pantalla aparecerá un mensaje que indica que el sistema operativo está activo y que pide que se ingrese la fecha del día (mes, día y año).
Una vez ingresada la fecha, hay que pulsar **RETURN**. Luego se debe escribir:

```
A>format
```

MSX-DOS pregunta a continuación cuál unidad de disco se va a utilizar:

```
Drive name? (A,B) ■
```

Si hay dos unidades de disco, se escribe la letra correspondiente a la unidad donde se colocará el disco a inicializar; en caso de contar con una sola, se escribe A.
Aparece luego el siguiente mensaje:

```
Strike a key when ready ■
```

que significa que hay que colocar el disco en la unidad correspondiente y apretar alguna tecla de la consola para que comience el proceso de inicialización.
Una vez completado el proceso, aparece el mensaje:

```
Format complete
```

lo que indica que el disco ya está listo para ser utilizado. Si se desea volver a utilizar MSX-Logo, hay que apagar la computadora y colocar nuevamente el cartucho Logo según las instrucciones dadas.

Nota: Si en lugar de "Format complete", aparece el mensaje "Write protected", significa que se está utilizando un disco de un solo lado. En este caso se debe comenzar de nuevo el procedimiento escribiendo **format/1** en lugar de **format**.

El proceso de inicialización de un disco debe hacerse solo una vez; si se utiliza el comando **format** con un disco que ya ha sido utilizado, este comando borrará todos los archivos que estén grabados en el disco.

El teclado de la Talent-MSX

El teclado de la computadora Talent-MSX y sus funciones son muy similares a los de una máquina de escribir. Además se encuentran disponibles gran cantidad de símbolos gráficos, así como letras usuales en idiomas distintos del castellano y aún letras griegas y símbolos matemáticos.
Sin embargo, algunas de las teclas son especiales para computadora y otras trabajan de forma ligeramente distinta a la habitual de una máquina de escribir.
Otras, finalmente, son ayudas para la corrección de los textos que se ingresan a la computadora.
En las explicaciones que siguen se hace referencia al cursor. El cursor es un

cuadrado luminoso que titila e indica el lugar de la pantalla donde aparecerá el próximo carácter que se escriba.

Teclas especiales o de control

RETURN Es una tecla grande ubicada a la derecha del teclado. Se llama también tecla de entrada, pues cuando se pulsa al finalizar una instrucción, indica a la computadora que la ejecute.

CTRL Se la llama "tecla de control". Permite que otras teclas, usadas junto con ella, realicen funciones especiales. Por ejemplo presionando **CTRL -A** el cursor se ubicará al principio de la línea en que se encuentra. (**Ctrl -A** significa que hay que apretar la tecla **CTRL** y, manteniéndola baja, pulsar la tecla **A**. Las funciones asignadas a la tecla **CTRL**, se usan con frecuencia en el modo Editor y están explicadas en detalle en la sección correspondiente al Editor Logo.

STOP Es la tecla de "parada". Cuando se la pulsa simultáneamente con la tecla **CTRL**, detiene de manera definitiva cualquier programa o procedimiento en ejecución. Si se la pulsa sola, la tecla **STOP** detiene el procedimiento de manera temporaria; tocando cualquier tecla la ejecución se reanuda.

Nota: las teclas **TAB** y **ESC** solo se usan en el Editor Logo. Ver el Anexo B, Teclas para edición de texto y figuras.

Teclas de símbolos

SHIFT Las teclas **SHIFT**, situadas a ambos lados del teclado alfabético, son teclas de mayúsculas. Si se presiona esta tecla y un número, se imprimirá el símbolo impreso en la parte superior de la tecla.

CAPS Presionando esta tecla se cambiará la impresión de teclas alfabéticas de mayúsculas a minúsculas y viceversa; sin embargo, para usar los símbolos indicados en la parte superior de las teclas, es necesario pulsar **SHIFT**.

GRAPH Es similar a **SHIFT**. Pulsándola junto con otra tecla, se imprime el símbolo gráfico indicado en el frente de la tecla.

CODE Es similar a **SHIFT** y a **GRAPH**. Presionando junto con otra tecla, se obtendrá un símbolo especial. Por ejemplo, apretando **CODE-N** obtenemos la letra ñ.

/
\ Se la utiliza para escribir una vocal acentuada. Para el acento ortográfico castellano se debe pulsar primero esta tecla junto con **SHIFT**, soltar ambas y luego presionar la vocal que queremos acentuar.

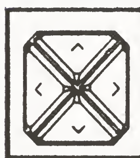


Teclas de corrección

BS Es la "tecla de retroceso y borrado". Presionándola se elimina el carácter ubicado a la izquierda del cursor. Todos los caracteres a la derecha del cursor se corren un espacio hacia la izquierda.

DEL Elimina el carácter ubicado bajo el cursor. Todos los caracteres a la derecha del cursor se corren un espacio hacia la izquierda.

Teclas de control del cursor



A la derecha del teclado hay un conjunto de cuatro teclas celestes que tienen forma triangular, y sirven para mover el cursor en las direcciones indicadas por ellas.
Se las utiliza comúnmente dentro del Editor, si bien, en el modo directo, las teclas de flecha izquierda y derecha pueden utilizarse para mover el cursor dentro de la línea que se está escribiendo.

El indicador Logo

? Cuando se enciende la computadora con el cartucho Logo, aparece un mensaje dando la bienvenida, y, en el renglón inferior, un signo de interrogación. Este signo es una invitación a escribir en la computadora. Mientras el símbolo se encuentre visible, significa que Logo espera el ingreso de instrucciones.

CAPÍTULO 2

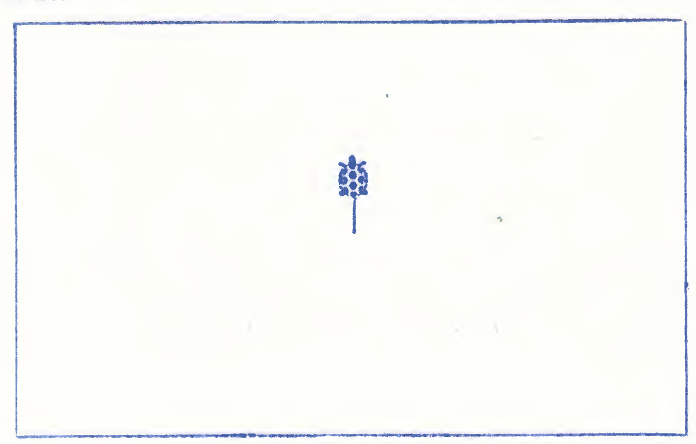
LOS GRAFICOS DE LA TORTUGA

Los primeros pasos

Cuando comienza Logo, una tortuga aparece siempre en el centro de la pantalla. La tortuga Logo * puede desplazarse por la pantalla, obedeciendo las órdenes que le damos en el teclado de la computadora. Al moverse, la tortuga deja una línea o trazo que va delineando un dibujo.

Para llevar la tortuga hacia adelante, escribamos:

?adelante 20.



Nota: las instrucciones de la computadora deben escribirse tal cual se indican. Si se comete algún error al escribir y no se ha pulsado **RETURN**, se puede corregir con las teclas de corrección (también llamadas de "edición"), las que se explican en el Capítulo 1 y en el Anexo B. La tecla **BS**, por ejemplo, borra el carácter ubicado a la izquierda del cursor.

** La designación de tortuga proviene del nombre que el neurofisiólogo Grey Walter dio a unos pequeños robots electromecánicos experimentados en Inglaterra a principio de 1960.*

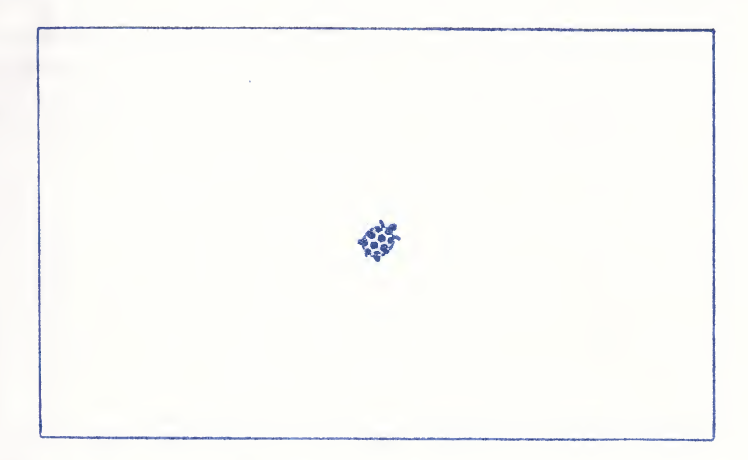
Al concluir la línea pulse la tecla **RETURN**. Inmediatamente, la tortuga avanzará 20 "pasos" hacia adelante. La orden **atrás** hace que se desplace la tortuga en el sentido opuesto al de **adelante**. Así,

?**atrás** 20 (pulse **RETURN** al concluir la línea)

llevará la tortuga a su posición inicial.

La tortuga se desplaza siempre en la dirección hacia adonde apunta su cabeza; esta dirección es la orientación o rumbo de la tortuga. Podemos modificarlo con las órdenes **derecha** o **izquierda**. Por ejemplo:

?**derecha** 40 (pulse **RETURN**)



La tortuga gira 40 grados hacia la derecha, sin moverse de su lugar; ahora apunta hacia una de las esquinas de la pantalla. Es importante notar que la tortuga no se ha desplazado, solo se ha modificado su rumbo.
La orden **izquierda** es similar a **derecha**, pero gira la tortuga en el sentido opuesto. Si escribimos ahora:

?**izquierda** 80

la tortuga gira y apunta hacia la otra esquina de la pantalla.
Con estas cuatro órdenes se puede hacer que la tortuga recorra toda la pantalla. También es posible usar sus abreviaturas (**ad, at, de** e **iz**). A medida que se mueve, la tortuga dejará un dibujo marcando el recorrido realizado.
Para borrarlo y volver la tortuga a la posición inicial, debemos escribir:

?**bg** (abreviatura de **b**orrar **g**ráficos)

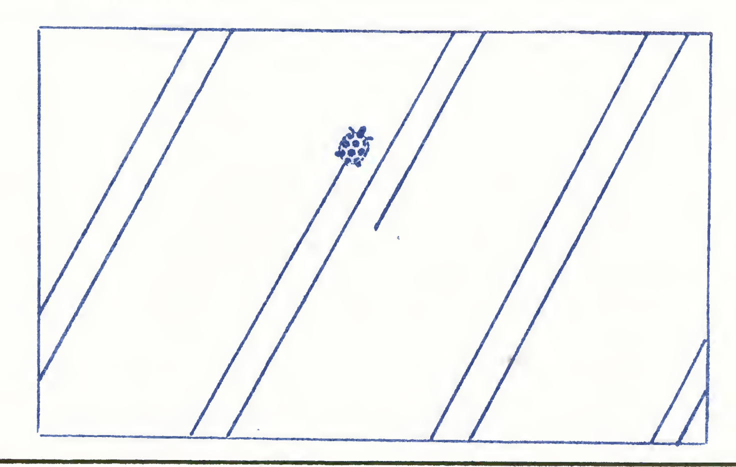
Esta orden borra el gráfico sin afectar el texto de la pantalla; a diferencia de las ya

vistas, no necesita ningún número a continuación.

Nota: si las órdenes no se escriben exactamente como Logo las espera, la computadora imprimirá un mensaje en la pantalla. En el Capítulo 9 'Mensajes de la Pantalla' encontrará una lista completa de los mensajes y una explicación de cuando ocurren.

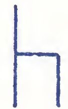
La tortuga, cuando sale por un borde de la pantalla, reaparece por el opuesto con la misma dirección y sentido. Este es el modo de la pantalla llamado **vuelta**; existe también el modo **ventana** que se explica en detalle en el Capítulo 8 Descripción de las Primitivas.
El modo **vuelta** nos permite realizar interesantes dibujos con pocas órdenes; por ejemplo:

?**derecha** 30 **adelante** 2000



Los primeros proyectos

Intentemos dibujar una silla como la del dibujo:



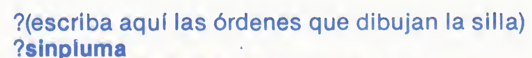
Por supuesto, existen muchas maneras de dibujarla y todas son igualmente buenas. Proponemos la siguiente:

adelante 40
atrás 20
derecha 90
adelante 16
derecha 90
adelante 20

La tortuga inicia el dibujo mirando hacia arriba en el extremo inferior de la pata izquierda; cuando termina se encuentra mirando hacia abajo en la pata derecha. Dibujemos ahora una sala, agregando una mesa.



La primera parte es la silla ya conocida; dibujar la mesa no es difícil, pero es necesario llevar la tortuga desde la silla hasta la mesa sin dibujar. Para ello utilizamos la orden **sinpluma**; **sinpluma** hace que la tortuga abandone su lápiz imaginario y se desplace sin dejar rastro.



?(escriba aquí las órdenes que dibujan la silla)
?**sinpluma**
?**izquierda** 90 **adelante** 30 **izquierda** 90

La tortuga se encuentra ahora correctamente ubicada para dibujar la mesa, pero, para que deje trazo, debemos dar nuevamente la orden **conpluma:**

?**conpluma**
?(escriba aquí las órdenes que dibujan la mesa)

sinpluma y **conpluma** tienen abreviaturas, que son respectivamente **sp** y **cp**.
Hay muchas órdenes relacionadas con los diseños de la tortuga; una lista completa se encuentra en el apartado Gráficos del Anexo G, Lista de primitivas agrupadas por función. Las más utilizadas son:

ot (ocultar la tortuga): hace invisible la figura de la tortuga.
mt (mostrar tortuga): hace visible la figura de la tortuga.
limpiar : borra el dibujo sin modificar la posición de la tortuga en la pantalla.
centro : lleva la tortuga al centro de la pantalla; si está conpluma, deja un trazo al moverse.
bp (borrar pantalla) : borra el dibujo y el texto de la pantalla.
bt (borrar texto) : borra el texto de la pantalla, sin afectar a los dibujos (inversamente a lo que sucede con **bg**).



Gráficos multicolores

Hasta ahora la tortuga ha dibujado con líneas blancas sobre fondo azul, pero en Logo hay 16 colores disponibles. Cualquiera de estos colores se pueden usar para modificar el color de la pluma o el del fondo de la pantalla.

En la siguiente tabla se indican los colores del MSX-Logo y el número que los identifica.

Tabla de colores

Número	Color	Número	Color
0	transparente	8	rosa
1	negro	9	rosa claro
2	verde medio	10	amarillo
3	verde claro	11	amarillo claro
4	azul oscuro	12	verde oscuro
5	azul claro	13	magenta
6	rojo	14	gris
7	celeste	15	blanco

Nota: la tortuga no dejará trazo visible si el color de la pluma es transparente (0) o igual que el color del fondo.

La orden **fcolorp** (fijar **color pluma**), seguida de un número, asigna a la pluma el color correspondiente al número. La orden **fcolorf** (fijar **color fondo**) es similar, pero modifica el color de la pantalla. Con la orden **cambiar.color** es posible modificar el color del texto o de los dibujos ya realizados. Por ejemplo:

?**cambiar.color** 15 1

pintará de negro (color 1) todos los textos y figuras de color blanco (15) que se encuentren en la pantalla.

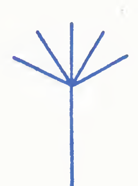
Las siguientes órdenes dibujan un ramillete multicolor en la pantalla:

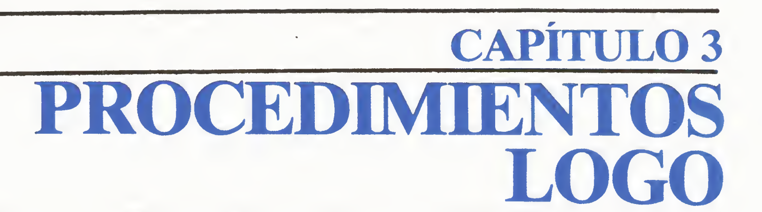
?**bp** (se borran todos los dibujos y el texto de la pantalla)
?**fcolorp** 2
?**at** 60 **ad** 60 (se dibuja el tallo en color verde medio)
?**iz** 60
?**fcolorp** 6 (se modifica el color de la pluma)
?**ad** 40 **at** 40 (se dibuja un pétalo de color rojo)
?**de** 30 (la tortuga gira un poco hacia la derecha)
?**fcolorp** 4 (se modifica nuevamente el color de la pluma)
?**ad** 40 **at** 40 (aparece un nuevo pétalo)
?**de** 30 (se repite el ciclo)

?**fcolorp** 8
?**ad** 40 **at** 40
?**de** 30
?**fcolorp** 7
?**ad** 40 **at** 40
?**de** 30
?**fcolorp** 10
?**ad** 40 **at** 40

Para resaltar la tarea realizada, ocultemos la tortuga y demos al fondo otro color:

?**ot fcolorf** 15





CAPÍTULO 3 PROCEDIMIENTOS LOGO

adelante, atrás, derecha, izquierda y las restantes órdenes vistas son palabras propias del lenguaje Logo, a las que en general nos referiremos como primitivas. Numerosas primitivas forman el vocabulario de Logo, pero su verdadera potencia reside en la posibilidad de definir nuevas palabras. De esta manera, el vocabulario disponible se puede extender en forma indefinida. *
En Logo las palabras son órdenes: impulsan a la tortuga a hacer algo. Inventar nuevas palabras significa, entonces, inventar nuevas órdenes (a las nuevas órdenes se las suele llamar procedimientos).

Un ejemplo, la silla

Enseñemos a Logo la palabra "silla". Cada vez que escribamos silla, la tortuga deberá ejecutar la secuencia de órdenes ya conocidas y dibujar la silla en la pantalla.
La palabra clave **para** se utiliza para indicar que se inicia la definición de un procedimiento. Escribamos

?**para** silla

y pulsemos **RETURN**. Las siguientes líneas aparecen en la pantalla:

para silla
fin

Pulsando **RETURN** nuevamente el cursor se ubica en una línea intermedia entre "**para** silla" y "**fin**". Definamos silla ingresando la secuencia de instrucciones ya vista:

para silla
ad 40
at 20
de 90
ad 20
de 90
ad 20
fin

* *Alas para la mente, Horacio C. Reggini, Bs. As. Galápago, 1982*

La pantalla muestra la estructura típica de la definición de un procedimiento: comienza con la palabra **para**, y, a continuación, el nombre o título del procedimiento, luego, la serie de instrucciones y, por último, la palabra clave, **fin**, que indica el final de la definición.

Nota: Logo se encuentra ahora en el modo Editor, al cual hemos ingresado con la orden **para**. En este modo, Logo escucha y toma nota de las definiciones de los procedimientos; las instrucciones que se le dan no son ejecutadas inmediatamente, a diferencia de como sucedía hasta ahora. Volvamos al modo normal (que se llama modo directo o "nivel superior"), pulsando la tecla **ESC**.

Cuando se pulse **ESC**, aparecerá el mensaje:

silla **definido**

Indicando que el procedimiento silla se ha agregado a las palabras que Logo reconoce. Hagamos una prueba:

?silla

Si le hemos dado las instrucciones correctas, la tortuga dibuja la silla. Ahora es muy fácil repetir el dibujo de la silla en cualquier lugar de la pantalla.

Nota: si hubo alguna equivocación al escribir las instrucciones de la silla, es muy sencillo corregirlas en el modo Editor. Se puede volver al modo Editor con la orden **para** seguida del nombre del procedimiento, en el caso anterior, **para** silla, o con otra primitiva, **editar** seguida del nombre del procedimiento precedido de comillas: **editar** "silla. **editar** puede también ir seguida por varios nombres de procedimientos, los que en tal caso deben estar entre corchetes. Ver el Anexo B, Teclas para edición de texto y figuras.

Procedimientos más poderosos

Otro aspecto fundamental de Logo es que en la definición de nuevos procedimientos se pueden utilizar no solo palabras primitivas sino también cualquier procedimiento nuevo. Por ejemplo, a partir de silla es muy sencillo hacer otro procedimiento que dibuje una platea o fila:

```
para fila
silla
sp iz 90 ad 15 iz 90 cp
silla
sp iz 90 ad 15 iz 90 cp
silla
sp iz 90 ad 15 iz 90 cp
silla
fin
```

fila **definido**
?fila



Se llama "subprocedimiento" al procedimiento que forma parte de otro; así, silla es un subprocedimiento de fila. fila es el "subprocedimiento" o procedimiento principal.
En Logo existe la palabra primitiva **repetir**, que nos permite simplificar el procedimiento anterior:

```
para fila
repetir 3 [silla sp iz 90 ad 15 iz 90 cp]
silla
fin
```

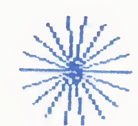
repetir es una primitiva que necesita dos datos (o "argumentos") y ambos de distinta naturaleza. El primero es un número y el segundo, una serie de instrucciones encerradas entre corchetes (en Logo, se llama **lista** cualquier cosa encerrada entre corchetes). **repetir** ejecuta la serie de instrucciones tantas veces como lo indica el número.

Podemos obtener hermosos diseños con repetir:

```
para pétalo
ad 20 at 20
fin

para flor
repetir 18 [pétalo iz 20]
fin
```

?flor



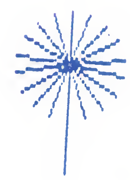
Otro ejemplo:

```
para palmera
tronco
copa
fin

para tronco
ad 40
fin

para copa
flor
fin

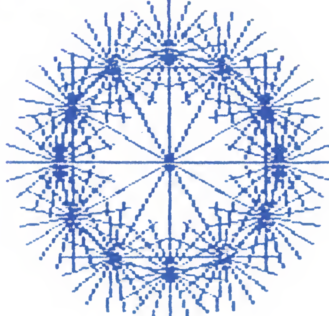
?palmera
```



O, por último:

```
para fantasía
repetir 10 [palmera at 40 de 36]
fin

?fantasía
```



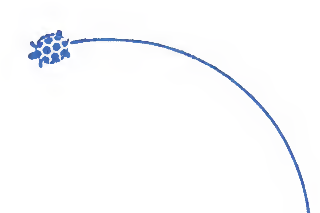
Utilizar las palabras propias para la definición de nuevos procedimientos es una idea poderosa. No sólo nos ahorra el esfuerzo físico de escribir las líneas, sino que simplifica la concepción y puesta a punto de los proyectos. Cada parte de una idea puede probarse, corregirse y modificarse en forma individual, permitiendo que concentremos nuestra atención sólo en "porciones de problema" de un tamaño adecuado.*

* *Ver Desafío a la Mente, Seymour Papert, Buenos Aires, Galápago, 1981.*

El círculo Logo

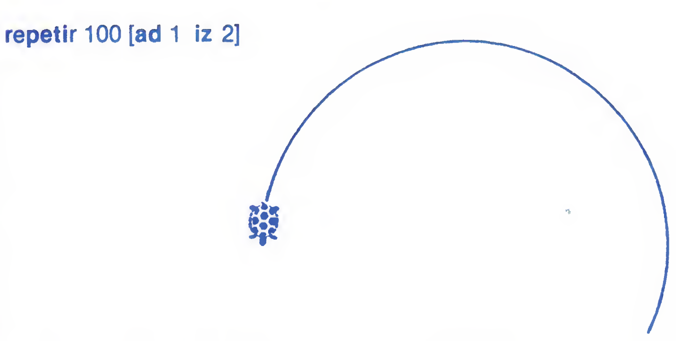
La tortuga se mueve solo con pasos rectos. ¿Cómo podemos enseñarle a hacer un círculo?
Una manera de solucionar el problema es "jugar a la tortuga". Pensemos como hacemos nosotros para dar una vuelta a la fuente circular de una plaza. Caminamos un poquito y giramos un poquito:

```
?repetir 50 [ad 1  iz 2]
```

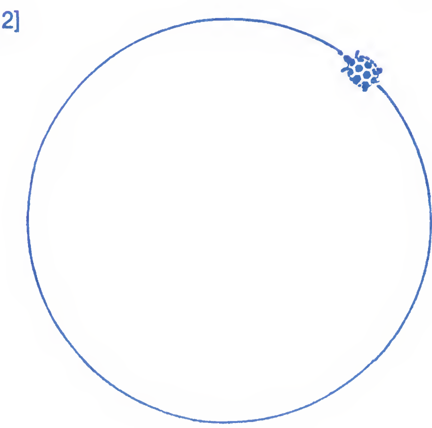


Estas órdenes dibujan un círculo incompleto. Si aumentamos el número de **repetir** se acercará cada vez más a un círculo.

```
repetir 100 [ad 1  iz 2]
```



```
?repetir 200 [ad 1  iz 2]
```



100 no alcanza y 200 es demasiado. El número correcto es 180 que hace que la tortuga gire 180 × 2 = 360 grados. Este es el Teorema de la Vuelta Completa de la Tortuga: si la tortuga realiza una vuelta completa, la suma de todos los giros es 360 grados.

Procedimientos con argumento

Hemos visto que algunas órdenes necesitan de un valor para poder ejecutarse. Por ejemplo, **adelante** necesita de un número que le indique a la tortuga cuántos pasos debe moverse. Este valor que sigue a una orden, se denomina el argumento o dato de la orden.
Las palabras que nosotros definimos o sea los procedimientos, pueden también incluir argumentos, es decir, valores variables elegidos a voluntad.

Propongámonos crear un procedimiento que dibuje cuadrados cuyo lado sea de longitud arbitraria. Pensemos, en primer lugar, cómo sería un procedimiento que realice cuadrados de lados de longitud 10.

```
para cuadrado
repetir 4 [ad 10 de 90]
fin
```

```
?cuadrado
```

Un procedimiento cuadrado con lado variable debería ser algo así como:

```
para cuadrado (valor del lado)
repetir 4 [ad (valor del lado) de 90]
fin
```

donde "valor del lado" valdría 10, 72, 32, etc... de acuerdo a lo que deseemos.
En Logo, expresamos eso así:

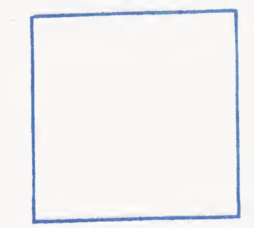
```
para cuadrado :longitud
repetir 4 [ad :longitud de 90]
fin
```

(longitud debe llevar dos puntos pegados)

Hemos reemplazado el número 10, que indicaba cuánto debía avanzar la tortuga, con una palabra; esta misma palabra se encuentra también a continuación de **para** cuadrado.



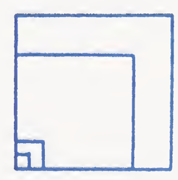
```
?cuadrado 70
```



Cuando ordenamos cuadrado 70, el número 70, argumento de la orden cuadrado, sustituye a :longitud, y cada vez que Logo encuentra la palabra :longitud en el procedimiento, recuerda que a ella se le ha asignado el valor 70.

Veamos así cómo podemos hacer cuadrados de distintos lados:

```
?cuadrado 10
?cuadrado 5
?cuadrado 60
?cuadrado 45
```



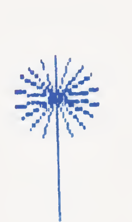
Construyamos ahora palmeras de distintos altos:

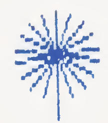
```
para tronco :alto
ad :alto
fin
```

```
para palmera :alto
tronco :alto
copa
fin
```

Para que el dibujo de la palmera entre en la pantalla, llevemos la tortuga hacia abajo con **sp at 65 cp.**

```
?sp at 65 cp palmera 60
?bg sp at 65 cp palmera 100
?bg sp at 65 cp palmera 30
```





Guardar y recordar

En este capítulo hemos enseñado muchas palabras nuevas a **Logo**, pero solo las puede recordar mientras la computadora permanezca encendida: si ahora se desconectara la máquina, todo el trabajo realizado se perdería. Para conservar los procedimientos y poder volverlos a usar es necesario grabarlos en un disco o en una cinta de casete.

Cómo usar la unidad de disco

En primer lugar, hay que conectar la unidad de disco en la forma indicada en el **Capítulo 1** de este manual (o en el de la unidad de disco), en donde también se indica cómo "inicializar" o "formatear" un disco para su uso con la computadora. Todo disco que se use debe estar inicializado de acuerdo a lo indicado allí.
La orden **guardar** hace que Logo copie en un disco el contenido del espacio de trabajo (es decir, la porción de memoria donde guarda todo lo que ha aprendido).
La orden **guardar** debe ir seguida de un nombre que identifica la información almacenada dentro del disco; el nombre del archivo debe tener menos de 8 letras precedidas por comillas (no van comillas al final). Por ejemplo:

?**guardar** "diseños

crea en el disco un archivo llamado "diseños, y guarda en él todos los procedimientos.
Es conveniente usar como nombre del archivo una palabra que describa lo mejor posible la información guardada.
Cuando se pulse **RETURN** se iluminará la lucecita de la unidad de disco; esta señal indica que la unidad está grabando o leyendo y mientras está encendida NO SE DEBE QUITAR el disco de la unidad. Cuando finaliza la grabación, aparece el signo de interrogación que indica que Logo espera nuevas órdenes.

Nota: Logo no admitirá grabar en el disco si el nombre del nuevo archivo coincide con alguno de los ya existentes: si se desea mantener el nombre, será necesario borrar del disco el archivo anterior del mismo nombre. Ver el uso de las primitivas **boarchivo** y **dir** en el Capítulo 8, Descripción de las primitivas.

El contenido de un archivo se trae del disco al espacio de trabajo, con la orden **recordar**, seguida por el nombre del archivo. Por ejemplo:

?recordar "diseño

Nuevamente se encenderá la luz de uso de la unidad de disco y, además, Logo imprimirá una serie de mensajes en la pantalla, a medida que define los procedimientos que contenía el archivo.

Cómo usar el grabador de cinta

En primer lugar, hay que conectar el grabador de la manera indicada en el Capítulo 1 de este manual.
La orden **guardac** hace que Logo grabe en una cinta el contenido del espacio de trabajo (es decir, la porción de memoria donde Logo guarda todo lo que aprendió). **guardac** debe ir seguida por un nombre, precedido de comillas, que identifica la información almacenada. El nombre no debe tener más de 6 letras.

?guardarc "tramas

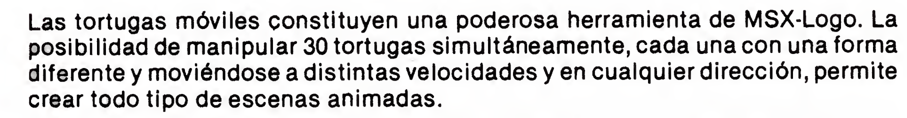
Las teclas del grabador deben estar en la posición de GRABACION. Si existe la conexión para REM (control remoto), la computadora pondrá en marcha y detendrá automáticamente el motor del grabador; si esta conexión no existe, se debe poner en marcha el grabador manualmente —siempre en modo GRABACION— antes de pulsar **RETURN**.
Una vez concluida la grabación, reaparece el signo de interrogación que indica que Logo espera nuevas órdenes (si la detención del grabador no es automática se lo debe detener ahora).
Para recordar un archivo, se debe ubicar la cinta en una posición previa a la iniciación del archivo, y pulsar la tecla de REPRODUCCION del grabador. A continuacún, se da la orden **recordarc** seguida del nombre del archivo:

?recordarc "tramas

Si existe la conexión REM, la computadora pondrá en marcha y detendrá automáticamente el motor del grabador; si no existe, todo esto se deberá hacer manualmente.

CAPÍTULO 4
LAS TORTUGAS MÓVILES

MSX-Logo posee la capacidad de manipular más de una tortuga (o "actor") simultáneamente. Cada tortuga puede cambiar su figura, su color o su movimiento de manera rápida y simple. Es posible activar en total treinta tortugas y cada una de ellas puede ser un pájaro, un auto u otra forma cualquiera elegida. Todas ellas dejan trazo al moverse, es decir que varias tortugas pueden dibujar simultáneamente.



Las tortugas móviles constituyen una poderosa herramienta de MSX-Logo. La posibilidad de manipular 30 tortugas simultáneamente, cada una con una forma diferente y moviéndose a distintas velocidades y en cualquier dirección, permite crear todo tipo de escenas animadas.

La tortuga, su disfraz y su velocidad

Cuando comenzamos a operar con MSX-Logo, la tortuga que aparece es la número 0 y tiene la figura de la tortuga. Si queremos que se disfrace con otra figura, decimos por ejemplo:

```
?ffig 4
```

que significa **f**ijar **fig**ura; ordena a la tortuga 0 "ponerse la figura 4". Esta figura corresponde a un camión. Existe un vestuario de numerosas figuras o disfraces que una tortuga puede utilizar. En el Anexo A, se encuentra la Tabla de figuras disponibles.

Las figuras de la 36 a la 59 son las correspondientes a la silueta habitual de la tortuga con diversas orientaciones.

Cualquiera de las figuras predefinidas en MSX-Logo pueden modificarse fácilmente por medio de la orden **edfig** (**ed**itar **fig**ura). La orden **edfig** seguida del número de una figura hace aparecer en la pantalla un cuadriculado donde se ve la figura correspondiente en un tamaño mucho más grande. La forma de la figura se cambia coloreando o no coloreando los cuadraditos. La tecla espaciadora cambia el estado de un cuadradito, de coloreado a transparente o viceversa; actúa sobre el cuadradito donde se encuentra el cursor titilante. Las teclas de control



del cursor (las celestes ubicadas a la derecha del teclado) sirven para situar el cursor en cualquier cuadradito que se desee modificar. Ver el Anexo B, Teclas para edición de texto y figuras para una explicación completa del uso del Editor de figuras.

Para dar por terminada la definición de una figura, salimos del Editor de figuras pulsando la tecla **ESC**.

La tortuga se puede mover permanentemente, es decir, es posible conferirle velocidad a la tortuga. La orden **fvel** (fijar **vel**ocidad) seguida por un número hace que la tortuga se mueva en la dirección que tiene a una velocidad igual al número dado. Dicho número puede variar entre – 128 y 128. Si le damos un número negativo, la tortuga se moverá hacia atrás.

Probemos con:

```
?fvel 50
```

La tortuga se mueve hacia adelante con la velocidad indicada y seguirá haciéndolo eternamente hasta que la detengamos con la orden **fvel** 0.
La tortuga dejará un rastro si está **conpluma**; el color del trazo puede variarse con la orden **fcolorp** seguida del número del color elegido, o, si ordenamos **sinpluma**, continuará moviéndose sin dibujar.

Cómo hablar a otras tortugas

Hasta ahora siempre nos hemos comunicado con una tortuga sola, la número 0 (cero). En MSX-Logo están a nuestra disposición 30 tortugas a las cuales podemos dar órdenes.

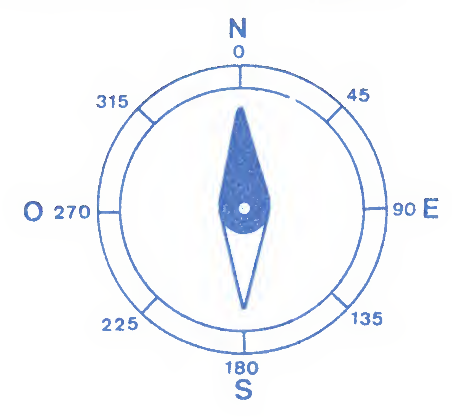
A fin de hablar o **activar** a una tortuga en particular, por ejemplo la 1, decimos:

```
?activar 1
```

Todas las órdenes que escribamos a continuación serán escuchadas y ejecutadas exclusivamente por la tortuga 1. Supongamos que queremos convertirla en un perro amarillo que corra hacia la derecha. Escribimos:

```
?activar 1
?mt
?ffig 3
?fcolor 10
?frumbo 90
?fvel 6
```

El perro correrá para siempre en la pantalla hasta tanto le ordenemos otra cosa a la tortuga 1.
En las órdenes anteriores hay dos nuevas primitivas. Una es **fcolor**, que da al disfraz de la tortuga el color indicado por el número dato. (Los números de los colores de las figuras son los mismos que ya vimos para la pluma y el fondo).
Otra orden nueva es **frumbo** (fijar **rumbo**). **frumbo** necesita un número de grados (de 0 a 360), que indica la orientación o rumbo que se desea para la tortuga.
El rumbo se mide desde la vertical (rumbo 0) y en el sentido horario. Si imaginamos que en el centro de la pantalla hay una brújula con el norte hacia arriba, los rumbos coinciden con las indicaciones de su cuadrante:



Nota: cuando comienza Logo, todas las tortugas están en el centro de la pantalla e invisibles. Para ver una tortuga, por ejemplo la 4, es necesario moverla o correr las tortugas superiores y ordenar **mt.**

Cómo hablar con muchas tortugas al mismo tiempo

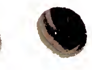
Es posible que en alguna ocasión deseemos hablar o comunicarnos con varias tortugas al mismo tiempo. Podemos hacerlo, escribiendo **activar** seguido de una lista formada por los números de las tortugas a las cuales queremos dirigirnos. Si deseamos hablarles a todas simultáneamente, escribimos:

```
?activar todas
```

todas es una primitiva que responde una lista con los números de todas las tortugas, es decir, de 0 a 29.

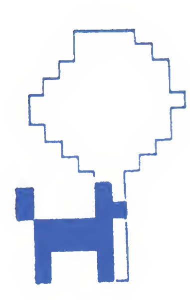
Las tortugas pueden disfrazarse con las figuras existentes; una vez que se ha definido una figura, cualquier tortuga, o todas, pueden adoptarla.

Si a varias tortugas se le dan las mismas instrucciones y parten del mismo lugar, parecerá que las cumple una sola tortuga, pues en general sólo es visible una de ellas (las otras no se pueden ver porque están debajo de la primera). Los números más bajos toman precedencia sobre los números más altos: si una o más tortugas se superponen, se ve la tortuga cuyo número es el más bajo. Es como si cada tortuga se moviera en planos distintos, donde el cero (0) es el plano superior, el 14 el central y el 29 el inferior.



Por ejemplo, una tortuga con forma de perrito puede pararse por delante de una tortuga disfrazada de árbol, siempre que la primera tenga un número más bajo que la segunda:

?**activar** 0
?**ffig** 14
?**activar** 1
?**mt color** 3



Se ha definido un perrito en la figura 14 y un árbol en la 15.

Nota: en la implementación actual solo 4 tortugas puede ser visibles simultáneamente en la misma línea; las de mayor número desaparecen.

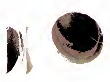
Diseños con varias tortugas

Otra manera de realizar figuras es hacer que varias tortugas dibujen simultáneamente, luego de haber partido de diversos lugares o con orientaciones diferentes. Por ejemplo:

?**activar** 0 **iz** 45
?**activar** 1 **de**135 **mt**
?**activar** [0 1]
?**fvel** 20 **repetir** 6[**esperar** 40 **iz** 60]
?**fvel** 0

Las tortugas 0 y 1 avanzarán simultáneamente en direcciones opuestas y dibujarán un hexágono cada una.
Con la orden **esperar** introducimos una pausa en la ejecución de las primitivas o los procedimientos por parte de Logo; la duración de la pausa es proporcional al número que sigue a **esperar**.
También se puede utilizar la orden **cada. cada**, seguida por una lista de instrucciones, impone esas instrucciones a cada una de las tortugas que se encuentran activas. Por ejemplo, si se escribe:

?**activar** [0 1 2]
?**cp mt**
?**cada** [**bp ffig** 1 **fcolor** 6 **esperar** 30
repetir 6[cuadrado **de** 60] **esperar** 60 **ot**]
la tortuga 0 aparecerá como un pequeño corazón rojo que obedecerá las órdenes




consignadas. Luego de un tiempo desaparece. A continuación, hará lo mismo la tortuga 1 y luego la 2.
Otra orden, **quién**, nos permite preguntar y usar el número o los números de las tortugas que están activas. Si ordenamos:

?**activar** 9
?**esc quién**

aparecerá en la pantalla el número 9 que es el número de la tortuga activa. El número dado como respuesta por la orden **quién** puede utilizarse en otras órdenes. Por ejemplo:

?**activar** 2
?**ffig quién**
?**fcolor quién**
?**mt**
?**frumbo** 10 * **quién**
? **fvel** 10 + **quién**

proporcionará a la tortuga 2 la figura del gato, color verde medio, rumbo 20 (10*2) y la velocidad 12 (10 + 2).
El empleo de la orden **quién** dentro de una lista de instrucciones gobernada por una orden **cada**, ofrece un medio para hacer que cada tortuga realice acciones diferentes. Por ejemplo:

para explosiones
activar todas
sp mt
cada [**ffig quién fcolor quién frumbo** 12 * **quién fvel** 8]
esperar 100 bp
explosiones
fin

El procedimiento explosiones hace que cada tortuga se vista con la figura que tiene su mismo número, se coloree con el color que tiene su mismo número, apunta cada tortuga hacia el rumbo que resulta de multiplicar 12 por su número y, finalmente da a cada tortuga la velocidad 8. Luego de esperar un tiempo y de llevar todas las tortugas al centro, recomienza todo el ciclo. Para detenerlo, es necesario pulsar **CTRL** - **STOP**.

Nota: explosiones es un ejemplo de "recursión", pues el procedimiento descripto incluye en su definición su propio nombre. Tal circunstancia se denomina "recursión", que significa "correr de nuevo" y es una de las ideas más poderosas de Logo. Usando la recursión es posible realizar procedimientos de pocas instrucciones que ejecuten acciones muy complejas. Ver **Alas para la mente**, Capítulos 12, 14 y 21.
Para volver a MSX-Logo a las condiciones habituales ordenamos:

?**rg** (restaurar gráficos)

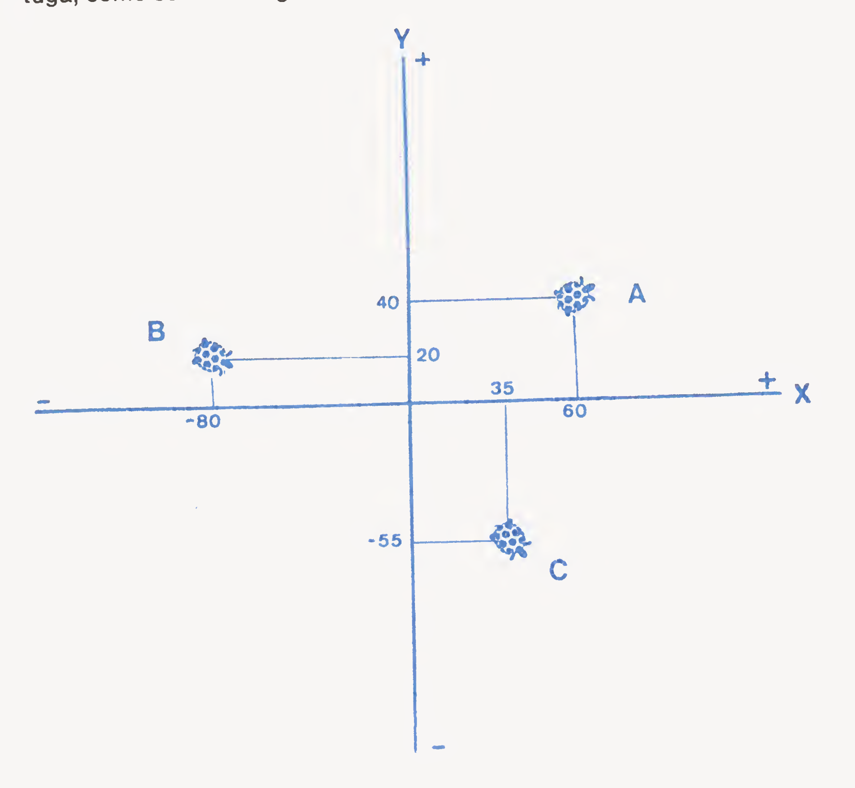
Quién, cómo, dónde

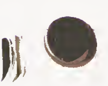
La primitiva **quién** también es útil cuando, luego de haber trabajado con varias tortugas, no recordamos cuál de ellas es la activa. El MSX-Logo existen otras órdenes que cumplen funciones similares y nos suministran información sobre las tortugas:

fig: responde con el número de figura de la tortuga activa.
color: responde con el número del color de la tortuga activa.
colorp, (color, pluma): responde con el número del color del trazo de la tortuga activa.
colorf (color fondo): responde con el número del color del fondo de la pantalla
rumbo: responde con el rumbo de la tortuga activa.
velocidad: responde con la velocidad de la tortuga activa.

Nota: si se ha activado un grupo de tortugas, estas órdenes responden con los valores correspondientes a la primera tortuga del grupo.

Hay tres primitivas que nos informan sobre la ubicación de la tortuga en la pantalla. Estas primitivas nos responden en función de las coordenadas x e y de la tortuga, como se ve en el gráfico:





En el gráfico se muestran dos ejes: el de las coordenadas x y el de las y; donde se cruzan los ejes, ambas coordenadas valen cero y es el centro de la pantalla. La tortuga A se puede ubicar diciendo que está a 60 pasos del centro hacia la derecha (zona de coordenadas xpositivas)y 40 pasos hacia arriba (zona de coordenadas y positivas). Observe que los ejes dividen la pantalla en cuatro zonas o cuadrantes, que se distinguen por los signos que tienen las coordenadas. Así, por ejemplo, en el segundo cuadrante (donde está la tortuga B), las coordenadas y son positivas (es decir que corresponden a la zona que está por encima del centro) y las coordenadas x son negativas (es decir, que corresponden a la zona situada a la izquierda del centro).
Las primitivas **coorx** y **coory** nos responden con el valor de la correspondiente coordenada de la tortuga activa. La orden **pos** nos responde con una lista cuyo primer elemento es el valor de la coordenada x, y el segundo, el valor de la coordenada y.

Nota: Ver en el Capítulo 8, Descripción de las primitivas, el funcionamiento de las órdenes **fx, fy** y **fpos**.

Pintar y sombrear

Las tortugas pueden colorear regiones completas de la pantalla, ya sea repitiendo una figura o pintando totalmente la región indicada con el color elegido.
La orden **pintar** se utiliza para llenar con color una zona de la pantalla delimitada previamente por el trazo de la tortuga. El color que usa la orden **pintar** es el color de la pluma de la tortuga activa. Varias tortugas pueden pintar zonas diferentes de la pantalla con distintos colores al mismo tiempo. Ejemplo:

```
?activar [0 1] mt
?cada [frumbo quién * 180]
?ad 10 cuadrado
?sp de 45 ad 10
?cp fcolorp 10 pintar
```

sombrear es parecido a **pintar**:. Copia la figura de la tortuga activa hasta llenar el espacio dado. La figura se pinta con el color de la pluma de la tortuga (no con el de la tortuga). El funcionamiento de **sombrear** se puede probar con el ejemplo de **pintar**.

Demonios

La metáfora de un "demonio" expresa la idea de que la orden crea una entidad autónoma dentro del sistema de la computadora, la cual permanece latente hasta que ocurre un cierto suceso, y entonces, como un demonio, sale de un salto para realizar su acción *. A la palabra "demonio" le damos aquí un significado similar al que tiene en la clásica frase de la física "el demonio de Maxwell".

* *Ver Capítulo 4, "Lenguajes para computadoras y para personas",* ***Desafío a la mente,*** *Seymour Papert, op. cit.*

El demonio es un tipo especial de medio de programación; su funcionamiento es independiente de todo programa en ejecución.
En MSX-Logo hay dos tipos de demonios: **cuando** y **al.contacto**. Los demonios **cuando** actúan si ocurre un acontecimiento especial. El funcionamiento del demonio **cuando** se ilustra con la instrucción.

```
?cuando 1 [actuar]
```

que significa que siempre que ocurre el acontecimiento 1 (pulsación de una tecla), se debe ejecutar la acción actuar.

En esta versión MSX-Logo, los demonios **cuando** vigilan seis tipos de acontecimientos, numerados del 0 al 5, que se indican en la siguiente tabla:

Tabla de acontecimientos

Número	Acontecimiento
0	transcurre alrededor de un segundo
1	pulsación de una tecla
2	pulsación del botón del mando 1
3	pulsación del botón del mando 2
4	movimiento de la palanca del mando 1
5	movimiento de la palanca del mando 2

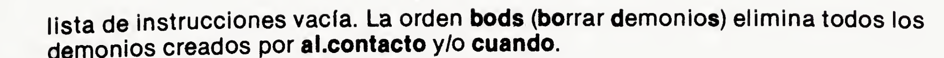
Cuando se produce el acontecimiento vigilado, el demonio interrumpe los procedimientos en ejecución y procesa una lista propia de instrucciones. Una vez que éstas se han ejecutado, el programa anterior continúa desde el punto de interrupción.
Por ejemplo:

```
?cuando 1 [ad 1]                  (hace que la tortuga avance un paso cada vez que
                                  se pulsa una tecla) O
?cuando 0 [sonido 0 440 10 10]    (se escucha un sonido cada vez que transcurre
                                  un segundo)
```

Los demonios **al.contacto** se establecen con una orden que requiere tres argumentos. Los dos primeros son números que designan a dos tortugas que el demonio vigila; el tercero es una lista de instrucciones que se cumple cuando las tortugas indicadas chocan. Por ejemplo:

```
?al.contacto 1 2 [de 90]     (cuando 1 y 2 se tocan, quién gira 90 a la derecha)
?al contacto 7 9 [disparar]  (cuando 7 y 9 se tocan, se realiza el procedimien-
                             to disparar)
```

Una vez que un demonio es activado, permanece alerta para siempre; se lo puede modificar redefiniéndolo con nuevas instrucciones o anularlo indicándole una

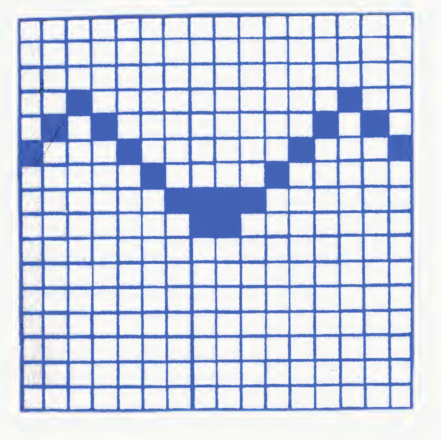


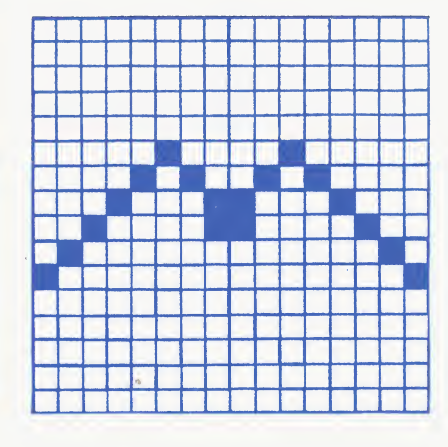
lista de instrucciones vacía. La orden **bods** (**bo**rrar **d**emonio**s**) elimina todos los demonios creados por **al.contacto** y/o **cuando**.

Primer ejemplo:

Dos tortugas que vuelan como pájaros

Como ejemplo de las posibilidades que proveen las tortugas múltiples para realizar escenas animadas, veremos cómo preparar una "película" que muestra una bandada de pájaros volando en la pantalla.
En primer lugar, definiremos las distintas formas que tiene un pájaro al batir sus alas, guardándolas en las figuras 12 y 13; las siguientes figuras sencillas bastarán:





Estas figuras se pueden crear fácilmente usando el Editor de figuras. A continuación escribiremos el procedimiento que ubica las tortugas en sus posiciones iniciales:

```
para bandada
activar [0 1 2 3 4 5 6]
ot bp
sp ad 18 de 60
cada [ad quién * 16]
de 30 fvel 30
mt
aletear
fin
```



El procedimiento bandada ubica primero las tortugas formando una bandada

que va desde el centro hacia la esquina superior derecha de la pantalla. El procedimiento aletear se ocupará de cambiar las figuras alternativamente entre sí:

```
para aletear
ffig 12 esperar 30
ffig 13 esperar 30
aletear
fin
```

aletear es un procedimiento recursivo. Cada vez que Logo llegue a su última instrucción, recomenzará el ciclo, y así indefinidamente.

Segundo ejemplo:

La tortuga y el perro



En el mundo de la fantasía de la pantalla, un perro huye en línea recta de una tortuga muy rápida que lo persigue. La tortuga, que no se encuentra sobre la línea,

recta que constituye la trayectoria del perro, lo persigue corriendo siempre en dirección hacia él. Nos planteamos los interrogantes siguientes: ¿cuál es la trayectoria de la tortuga?, ¿logrará escapar el perro fuera de la pantalla antes de que lo alcance la tortuga?

```
para caza :vperro :vtortuga
ubicar
activar 1 fvel :vperro
activar 0 fvel :vtortuga
al.contacto 0 1 [esc [perro apresado]]
orientar.tortuga
fin

para ubicar
ventana
activar 0 ffig 36 fcolor 1 mt
sp ad 60 iz 90
ad 110 iz 90
activar 1 ffig 3 fcolor 9 mt
sp de 180 ad 60
de 90 ad 110 de 180
fin

para orientar.tortuga
activar 0
frumbo hacia (pedir 1 [pos])
orientar.tortuga
fin
```

El procedimiento recursivo orientar.tortuga, apunta una y otra vez la tortuga hacia el perro; con ese fin utiliza la primitiva **hacia**, que responde con el rumbo requerido para mirar hacia un punto determinado de la pantalla. Por ejemplo:

```
?frumbo hacia [0 0]
```

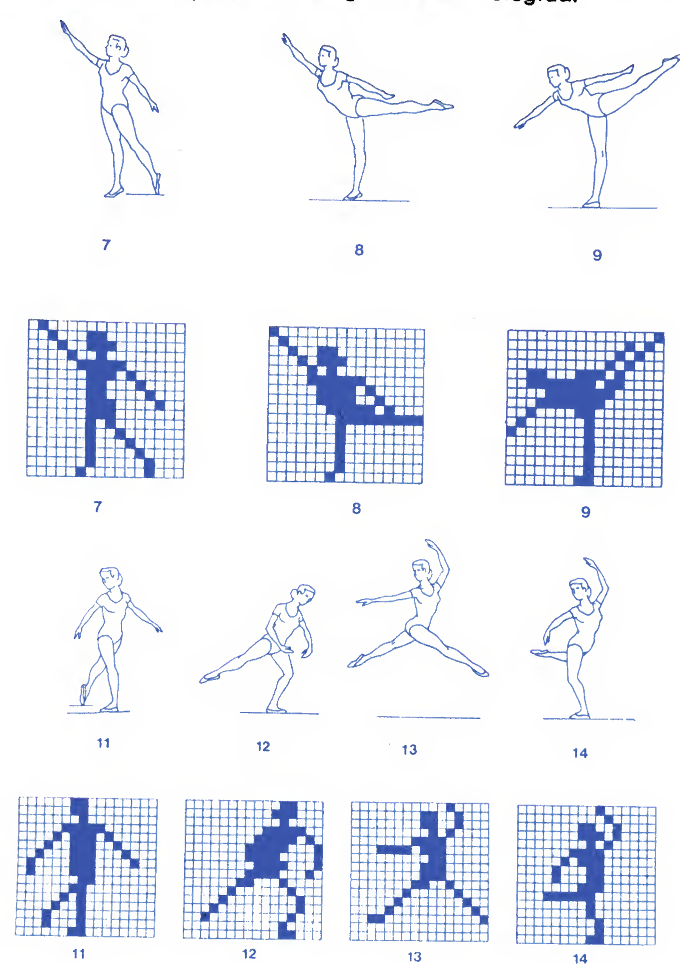
hará que la tortuga mire hacia el punto de coordenada x 0, y coordenada y también cero, es decir, el centro de la pantalla.

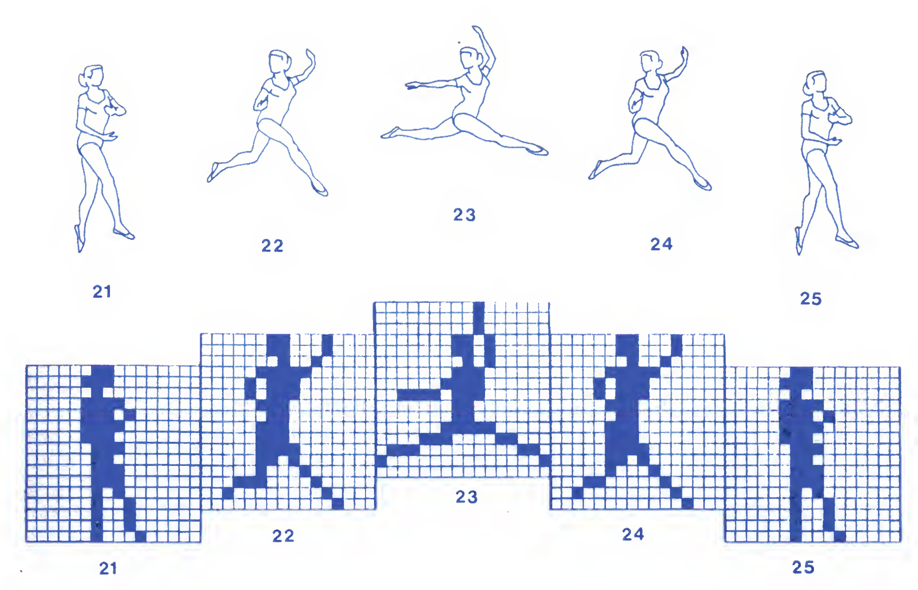
En orientar.tortuga, el dato de **hacia** es la posición de la tortuga 1 (el perro). Para obtenerla, debemos activar momentáneamente la tortuga 1; de otra manera, **pos** nos respondería la posición de la tortuga activa, que es la 0. Para ello se usa **pedir**, que es un **activar** temporario. Tiene vigencia solo para las tortugas indicadas a continuación, las que deben cumplir la lista de instrucciones que siguen. Una vez cumplidas, la o las tortugas activas continúan siendo las mismas que antes de ordenar **pedir**.

Tercer ejemplo

El ballet

La reproducción de una figura real en la cuadrícula de 16 × 16 de las figuras requiere un poco de habilidad para esquematizar los rasgos distintivos del objeto.
Las figuras siguientes ejemplifican la creación, a partir de unos dibujos simples, de los diseños necesarios para representar algunas posiciones del ballet clásico.
Las figuras 7, 8, 9 corresponden a las posiciones del primer *arabesque á terre*, primer *arabesque* y primer *arabesque penchée*.
Las 11, 12, 13 y 14 corresponden al *grand jeté*; ídem las 21, 22, 23, 24 y 25.
El lector podrá componer algunos movimientos con las figuras presentadas o con otras, modificadas convenientemente según la danza elegida.





Cuarto ejemplo:

Las cuatro tortugas

Cuatro tortugas amigas se encuentran en un momento dado ocupando los cuatro vértices de una plaza. Cada una de ellas comienza a caminar dirigiéndose hacia la que ocupa el vértice siguiente en un orden cíclico (0 va hacia 1, 1 hacia 2, 2 hacia 3, 3 hacia 4), de modo que las cuatro trayectorias son similares.
¿Qué forma tendrán esas trayectorias?

```
para ubicar
activar [0 1 2 3]
fcolor 6 mt sp
activar 0
de 90 camino
activar 1
camino
activar 2
iz 90 camino
activar 3
iz 180 camino
fin
```

```
para trayectoria
activar 0
frumbo hacia (pedir 1 [pos])
activar 1
frumbo hacia (pedir 2 [pos])
activar 2
frumbo hacia (pedir 3 [pos])
activar 3
frumbo hacia (pedir 0 [pos])
trayectoria
fin

para camino
repetir 2 [ad 60 iz 90]
fin

para cuatro.tortugas
ubicar
activar [0 1 2 3]
cp fvel 1
trayectoria
fin
```

¿Qué ocurrirá cuando las amigas se encuentren?



En Logo se manipulan varios tipos de "cosas" u "objetos". Hasta ahora, por ejemplo, hemos estado manejando números para realizar dibujos o para dar movimiento a las tortugas, para indicar el tamaño del tronco de la palmera o el número de veces que hay que repetir una lista de instrucciones.
Pero también es posible manipular palabras; para Logo, una palabra es una sucesión cualquiera de símbolos (letras, números u otros) con la única excepción del espacio, que se usa, justamente, para separar una palabra de otra. Ya hemos usado algunas palabras: el nombre de un procedimiento o el nombre de una variable en los procedimientos con argumento.
Por último, podemos manipular "listas". Una lista es un conjunto de números, palabras u otras listas, todo encerrado entre corchetes, por ejemplo:

```
[éste es el ejemplo número 1 de listas]
```

es una lista. Otros ejemplos son la lista de instrucciones de **repetir** o la lista de números que escribimos cuando activamos varias tortugas simultáneamente.
La palabra primitiva **escribir** nos permite ver los objetos Logo. Por ejemplo:

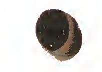
```
?escribir 10
10
?escribir "hola
hola
?escribir [buenos días]
buenos días
```

Obsérvese que la palabra "hola lleva comillas adelante, de manera que no se confunda con el nombre de un procedimiento o primitiva; además, **escribir** no muestra los corchetes exteriores de las listas.

Nota: para Logo, los números son también palabras pero de un tipo especial. Están formadas solamente por dígitos y pueden prescindir de las comillas iniciales.

Cómo guardar cosas en la memoria de Logo

Antes de empezar a trabajar con palabras y listas, veremos una nueva manera de almacenar cosas en la memoria de Logo.
Ya sabemos definir procedimientos, que hacen que Logo recuerde cómo reali-



zar ciertas acciones. Es posible también almacenar "cosas" como, por ejemplo, nombres de personas, números de teléfonos, direcciones, etc. Para ello se usa la orden **hacer**:

?**hacer** "dirección [Alem 1074 Buenos Aires]

En la línea anterior, hemos creado una variable de nombre "dirección" y hemos asignado como valor la lista [Alem 1074 Buenos Aires]. De esta manera se pueden almacenar datos u otros objetos para usarlos más tarde.
Para tener una idea de cómo es la memoria de la computadora, se puede hacer una analogía entre ella y una inmensa cajonera. Al comenzar Logo hay muchos cajones vacíos, disponibles para almacenar cosas. Cuando se ordena **hacer**, Logo pone sobre un cajón cualquiera una etiqueta con el nombre de la variable ("dirección), y en el interior coloca el valor ([Alem 1074 Buenos Aires]). El contenido del cajón se puede ver con **escribir**:

?**escribir** :dirección
Alem 1074 Buenos Aires

No hace falta saber dónde se encuentra el cajón; sólo es necesario recordar su nombre o etiqueta.

Nota: los dos puntos (:) adelante de la palabra "dirección" son muy importantes para Logo, pues indican que el que sigue es el nombre de una variable y que se le pide mostrar su contenido. No debe haber ningún espacio entre los dos puntos y la palabra. Si la variable no existe, aparecerá un mensaje de error.

Otra manera de llevar a cabo lo especificado anteriormente es mediante la primitiva **nombrar**:

nombrar [Alem 1074 Buenos Aires] "dirección

La orden **nombrar** designa a la lista [Alem 1074 Buenos Aires] con el nombre "dirección. Al igual que antes obtenemos:

?**escribir** :dirección
Alem 1074 Buenos Aires

Una mirada al interior de las listas y las palabras

Una lista puede contener numerosos elementos y es común que en un momento determinado solo necesitemos trabajar con uno de ellos en particular. Logo dispone de primitivas que nos permiten conocer el contenido de las listas.
La orden **primero** nos responde con el primer elemento de una lista:

?**escribir primero** [avión cohete barco automóvil]
avión

La orden **último** funciona de una manera similar:



?**escribir último** [avión cohete barco automóvil]
automóvil

Más poderosa es la orden ítem, que nos responde con un elemento de la lista elegido a voluntad:

?**escribir ítem** 3 [avión cohete barco automóvil]
barco

La orden **ítem** requiere dos argumentos: el primero es el número de orden del elemento buscado, el segundo argumento es la lista que se desea analizar.
Otras primitivas útiles son **menosprimero** y **menosúltimo. menosprimero** responde con la lista sin su primer elemento:

?**escribir menosprimero** [avión cohete barco automóvil]
cohete barco automóvil

La función de la orden **menosúltimo** es análoga:

?**escribir menosúltimo** [avión cohete barco automóvil]
avión cohete barco

Estas primitivas se pueden aplicar también a palabras:

?**escribir primero** "bueno
b
?**escribir menosúltimo** "bueno
buen

En Logo, también existe la lista vacía, es decir, aquélla que no contiene nada. Se indica por un par de corchetes: [ ]. La aplicación de **menosprimero** o **menosúltimo** a una lista de un solo elemento, produce la lista vacía.

?**menosprimero** [Logo]
No sé que hacer con []

La utilidad del uso del conjunto de **primero** y **menosprimero** — o **último** y **menosúltimo** — se hace evidente en los procedimientos que actúan sobre listas. Como ejemplo, definimos una orden que escriba los elementos de una lista en renglones sucesivos, en lugar de hacerlo en un mismo renglón como **escribir**. El primer paso será escribir el primero de la lista; luego habrá que escribir el primero de la lista menos el primero de sus elementos y así sucesivamente hasta que la lista esté vacía. El procedimiento siguiente es una solución del problema:

```
para encolumnar :lista
escribir primero :lista
encolumnar menosprimero :lista
fin
```




Este procedimiento actúa recursivamente sobre la lista y va "comiendo" progresivamente el primer elemento de la lista que va quedando. Obsérvese que luego de **escribir** el primer elemento, el segundo paso es simplemente aplicar el mismo procedimiento encolumnar a la nueva lista que resulta de quitar el primer elemento a la anterior.
Pero este procedimiento no está completo; falta la "condición de parada", que detenga el procedimiento cuando la lista esté vacía. Si ordenamos:

```
? encolumnar [este procedimiento no funciona bien]
este
procedimiento
no funciona
bien
primero no acepta [] como argumento en encolumnar
```

El mensaje nos dice que no se puede obtener el primer elemento de una lista vacía. Hagamos que el procedimiento pare cuando llegue a ese punto modificándolo de la siguiente manera:

```
para encolumnar :lista
si :lista = [] [parar]
escribir primero :lista
encolumnar menosprimero :lista
fin
```

Obsérvese la sintaxis de **si**; a continuación de la palabra **si**, se encuentra una prueba; en este caso, preguntamos si la lista está vacía. La prueba, o predicado, puede tener solamente dos valores: **cierto o falso**. Si el predicado es **falso**, Logo no hace nada; si es **cierto**, cumple las órdenes indicadas en la lista que sigue —en este caso, **parar**. La palabra primitiva **parar** hace que se detenga la ejecución del procedimiento.

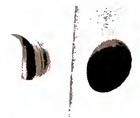
Nota: La orden **si** puede aceptar dos listas de instrucciones. Ver más adelante en Procedimientos interactivos o en el Capítulo 8, Descripción de las primitivas. Probemos:

```
?encolumnar [lo esencial es invisible a los ojos]
lo
esencial
es
invisible
a
los
ojos
```

Cómo inventar listas y palabras

Una lista se puede crear con la orden **hacer**:

```
?hacer "flores [azucenas rosas lirios]
```



Otra forma, muy común, es ampliar una lista existente. Hay varias maneras de hacerlo, por ejemplo, con la orden **frase**, que toma dos objetos y los devuelve juntos dentro de una nueva lista. Si ordenamos:
frase [azucenas rosas lirios] [siemprevivas margaritas]

se crea la lista [azucenas rosas lirios siemprevivas margaritas].
Los argumentos de **frase** pueden ser palabras o listas, pero el resultado es siempre una lista:

```
?mostrar frase "azucenas "margaritas
[azucenas margaritas]
?mostrar frase [azucenas rosas] "margaritas
[azucenas rosas margaritas]
```

Nota: mostrar es una primitiva cuya función es muy parecida a la de **escribir**; se diferencia en que muestra los corchetes exteriores de las listas que imprime. La primitiva **lista** es análoga a **frase**, pero, si alguno de sus argumentos es una lista, pasa a ser una "sublista" de la lista resultado:

```
?mostrar lista "azucenas "margaritas
[azucenas margaritas]

?mostrar lista [azucenas rosas] "margaritas
[ [ azucenas rosas ] margaritas ]
```

Otras primitivas que ingresan o entran objetos al principio o al final de una lista, son respectivamente **ponpri** (**pon**er **pri**mero) y **ponúlt** (**pon**er **últ**imo). Por ejemplo:

```
?escribir ponpri "caminante [ son tus huellas ]
caminante son tus huellas

?escribir ponúlt "caminante [ son tus huellas ]
son tus huellas caminante
```

El segundo dato de **ponpri** y de **ponúlt** debe ser siempre una lista, el primero, en cambio, puede ser una palabra o una lista. Si es una lista, ingresa en la segunda como sublista; así por ejemplo:

```
ponpri [ el camino ] [ y nada más ]
```

produce la lista:

```
[ [ el camino ] y nada más ]
```

Estas primitivas son válidas sólo para listas. Las palabras se unen mediante la orden **palabra:**

```
?escribir palabra "in "móvil
inmóvil
```

Las órdenes explicadas precisan en general dos argumentos, pero aceptarán

más si se encierra toda la expresión entre paréntesis. Por ejemplo:

```
escribir (palabra "in "móvil "es)
inmóviles
```

```
?mostrar (frase [ gentes ] [ que ] [bailan o ] [ juegan ])
[ gentes que bailan o juegan ]
```

Un cuento *

Con las órdenes vistas podemos formar oraciones simples. En el siguiente procedimiento, Logo escribe un cuento, inventando frases con elementos tomados al azar de varias listas dadas.
Para seleccionar un elemento cualquiera de una lista, se usa la orden **item** combinada con la orden **azar**, que responde con un número cualquiera menor que un límite dado. Así, si ordenamos:

```
?repetir 10 [ escribir azar 20 ]
```

aparecerán en la pantalla diez números elegidos al azar comprendidos entre 0 y 19. De la misma manera,

```
?escribir item (1 + azar 5) [ alfombra silla cortina sillón espejo ]
```

escribirá un elemento cualquiera de la lista.

Nota: obsérvese que **1 + azar 5** produce un número al azar comprendido entre 1 y 5; si no se hiciera así, **azar** puede responder 0, provocando un mensaje de error pues el **item 0** de una lista no existe.

El procedimiento cuento está dividido en dos partes o subprocedimientos. La primera parte define las listas que se van a utilizar.

```
para comenzar
hacer "sujetos [ senda noche luna planta luz mujer muñeca silla ]
hacer "adjetivos [hermosa alta brillante grande cansada verde roja dulce misteriosa alegre ]
hacer "verbos [ caminaba [ se asomaba ] sentía [ se encendía ] pensaba reía ]
hacer "lugares [ [en el jardín] [en su casa] aquí allí [durante el viaje] ]
fin
```

Nótese que dentro de las listas no hay solamente palabras sino también sublistas como [se encendía] o [en su casa].

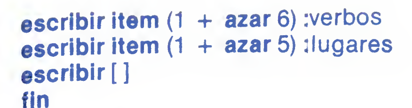
El subprocedimiento oración tiene a su cargo la creación de una oración al azar:

```
para oración
escribir "la
escribir item (1 + azar 8) :sujetos
escribir item (1 + azar 10) : adjetivos
```

* *Ver **Desafío a la Mente**. Cap. 2. 3ª. As. Galápago, 1981.*







```
escribir item (1 + azar 6) :verbos
escribir item (1 + azar 5) :lugares
escribir [ ]
fin
```

Cada elemento de la oración aparece en un renglón distinto para dar más claridad al programa, pero se podrían haber combinado en uno solo usando **frase**; el último **escribir** no muestra nada, simplemente deja un renglón en blanco.
El superprocedimiento cuento es:

```
para cuento
comenzar
repetir 4 [oración]
fin
```

Cada vez que se ordene cuento, aparecerán en la pantalla cuatro oraciones cualesquiera. Por ejemplo:

```
la
mujer
misteriosa
reía
en el jardín

la
planta
grande
se
asomaba
en su casa

la
luna
cansada
reía
allí

la
noche
hermosa
caminaba
en el jardín
```




Procedimientos interactivos

Logo permite escribir procedimientos que hagan preguntas al usuario, lean sus respuestas del teclado y reaccionen frente a ellas. Este tipo de procedimientos se llaman interactivos, pues requieren la intervención activa del usuario.



La orden **ll** (leer línea) hace que un procedimiento lea un mensaje que escribimos en el teclado. De esta manera, es posible simular una conversación con la computadora:

```
para conversar
escribir [buenos días]
escribir [cómo te llamas?]
hacer "nombre ll
escribir frase [bienvenido, ] :nombre
fin
```

Cuando se ejecuta **ll**; solamente el cuadrado luminoso aparece en la pantalla:

```
?conversar
buenos días
cómo te llamas?
```

Escriba su nombre y pulse **RETURN**, Logo le dará la bienvenida.
Es posible dar más variedad a conversar usando la orden **si**:

```
para conversar
escribir [buenos días]
escribir [cómo te llamas?]
hacer "nombre ll
escribir frase [ bienvenido, ] :nombre
escribir [qué color te gusta?]
si ll = [azul] [escribir [es un color muy lindo]] [escribir
[lástima, mi preferido es el azul ] ]
fin
```

Obsérvese que a continuación del predicado de **si** hay dos listas de instrucciones; si el predicado es **cierto**, se cumplirá la primera; si es **falso**, la segunda.

Nota: la palabra azul, en la comparación **ll** = [azul], debe estar dentro de una lista; se debe hacer así porque **ll** encierra lo que lee en una lista y, para Logo, la palabra "azul no es igual a la lista [azul].

Probemos conversar:

```
?conversar
buenos días
cómo te llamas?
Mario
bienvenido, Mario
qué color te gusta?
rojo
lástima, mi preferido es el azul
```



Lista de propiedades

Hemos visto cómo asignar un valor a un objeto Logo con la primitiva **hacer**. Por ejemplo:

```
?hacer "manzana "rojo
```

le asigna al objeto manzana el color rojo.

Las listas de propiedades permiten asignar más de un valor a un objeto; por ejemplo, se pueden asignar al objeto manzana las propiedades correspondientes a su sabor, forma y color.

Una lista de propiedades está siempre asociada a un objeto y está formada por uno o más pares de propiedades. El primer elemento del par es el nombre de la propiedad; el segundo, el valor que se le da.

Un ejemplo de una lista de propiedades asociada con el objeto libro puede ser:

```
[formato grande tema historia impresión [de lujo]].
```

Los nombres de las propiedades y sus valores pueden ser palabras o listas. Las cuatro primitivas que permiten la manipulación de las listas de propiedades son: **pprop, obprop, remprop** y **listaprops**.

Se usa **pprop** para asignar una lista de propiedades a un objeto; necesita tres argumentos: el primero es el nombre del objeto al cual se va a asociar la lista de propiedades; el segundo es el nombre de la propiedad y el tercero es el valor de la propiedad. El primer argumento de **pprop** debe ser una palabra; el segundo y el tercero pueden ser una palabra o una lista.

```
?pprop "María "teléfono [654123]
?pprop "María "calle [Arenales 1530]
?pprop "Juan "teléfono [262345]
?pprop "Juan "calle [México 145]
```

Para conocer el valor de una propiedad específica, hay que usar **obprop**:

```
?escribir obprop "María "teléfono
654123
```

Si no existe dicha propiedad, devuelve una lista vacía.

```
?mostrar obprop "María "edad
[]
```

La orden **listaprops** (**lista** de **prop**iedades) nos provee de una lista completa de las propiedades de un objeto y sus valores asociados. Por ejemplo:

?**mostrar listaprops** "María
[teléfono [654123] calle [Arenales 1530]]

?**mostrar listaprops** "Juan
[teléfono [262345] calle [México 145]]

Para borrar una propiedad específica y su valor de la lista de propiedades, se usa **remprop** (remover propiedad).

?**remprop** "María "calle
?**mostrar listaprops** "María
[teléfono [654123]]

Algo más sobre palabras y listas

Una primitiva importante para el manejo de palabras y listas es **cosa**, que responde el valor de un nombre. Por ejemplo:

?**hacer** "adjetivo "bueno
?**escribir cosa** "adjetivo
bueno

También puede lograrse la misma respuesta usando los dos puntos (:) de esta forma:

?**escribir** :adjetivo
bueno

Se manejan aquí conceptos lingüísticos de una manera muy clara y natural. El ejemplo anterior muestra cómo la sintaxis Logo diferencia gráficamente las dos partes del signo lingüístico — significado y significante:
"_______ indica el nombre que se le da a una cosa (significante)
:____ o **cosa** "____ indica el contenido o valor del nombre (significado).
Los ejemplos siguientes ilustrarán sobre las posibilidades de la orden **cosa:**

?**hacer** "auto "rueda
?**hacer** "rueda "neumático
?**hacer** "neumático "cámara
?**escribir** :auto
rueda
?**escribir cosa** "auto
rueda
?**escribir cosa cosa "rueda**
cámara
?**escribir cosa cosa cosa** "auto
cámara

Otro ejemplo:

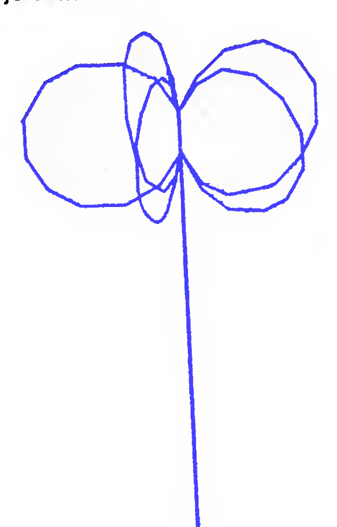
?**hacer** "herbívoros [vaca guanaco tortuga elefante]

?**hacer** "carnívoros [león perro puma lobo]
?**hacer** "animales [herbívoros carnívoros]
?**hacer** "vaca "pasto
?**escribir** :animales
herbívoros carnívoros
?**escribir cosa primero** :animales
vaca guanaco tortuga elefante
?**escribir cosa último** :animales
león perro puma lobo
?**escribir cosa primero cosa primero** :animales
pasto

CAPÍTULO 6

LOGO EN EL ESPACIO TRIDIMENSIONAL

Este capítulo conduce al lector hacia una aventura intelectual en el espacio, en la que se usa a la computadora como vehículo y a Logo como timón.*
En los capítulos anteriores hemos aprendido a realizar gráficos cada vez más complejos. Sin embargo, todos ellos son planos, ya que las órdenes aprendidas hasta ahora solo nos permiten mover la tortuga sobre la superficie del plano. Pero! ¿cómo hacer un dibujo como éste?



En este capítulo aprenderemos a manejar la tortuga tridimensional, aquella que puede escaparse del plano para dibujar figuras de tres dimensiones.

Nota: a fin de operar con las órdenes tridimensionales, se deben ingresar previamente en memoria los procedimientos guardados en el archivo TRI. Ver el Anexo D. Estos procedimientos no están disponibles en el cartucho de MSX-Logo, pero se pueden obtener en un casete adicional.

* *Ver **Ideas y formas**. Horacio C. Reggini, Bs. As., Galápago, 1985, Dist. Emecé.*

La tortuga tridimensional

Vamos a preparar la escena para la realización de un dibujo tridimensional. Escribamos:

?**tri**

Esta orden (abreviatura de **tri**dimensional), borra la pantalla y prepara la tortuga para dibujar una figura en el espacio.
¿Qué ha pasado con la tortuga? Hay que tener en cuenta que **tri** hace que la tortuga clásica desaparezca de la pantalla y pone en acción la tortuga tridimensional. Su figura no aparece en la pantalla; si deseamos conocer su posición tenemos que escribir:

?**triada**

En el centro de la pantalla aparecen tres ejes que componen la **triada**. ¿Qué indican estos ejes? Uno de ellos apunta en este momento hacia arriba; es el eje longitudinal de la tortuga e indica hacia dónde mira la tortuga tridimensional y muestra, por lo tanto, la dirección de su movimiento de traslación. Vemos otro eje, que va del centro de la pantalla hacia la izquierda; este eje es perpendicular al anterior y transversal a la tortuga; se halla en su plano medio, paralelo a sus patas y a su caparazón, y está orientado hacia la izquierda de la tortuga. Hay un tercer eje, que es perpendicular al longitudinal y al transversal y que apunta desde el centro de la tortuga hacia su caparazón. Como ahora es perpendicular a la pantalla no es posible verlo.
Estos tres ejes están fijos al centro de la tortuga y viajan con ella; de esta manera nos permiten conocer la posición y la orientación de la tortuga cuando se mueve en el espacio.
Si ordenamos nuevamente **triada,** ésta desaparece de la pantalla:

? **triada**

Cómo mover la tortuga tridimensional

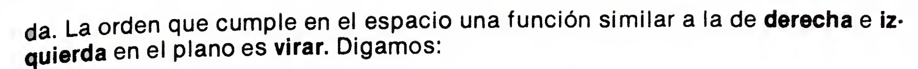
Al igual que la tortuga clásica, la tortuga tridimensional se desplaza por el espacio dejando un rastro. La orden que hace que la tortuga se mueva en la dirección hacia la que apunta es **andar.** Digamos:

?**andar** 50

La tortuga se ha movido 50 pasos hacia adelante. ¿Cómo hacer que la tortuga camine 50 pasos hacia atrás en el espacio para volver a su posición inicial? Digamos:

?**andar** —50

Tenga en cuenta que la orden **andar,** que mueve la tortuga en el espacio, se corresponde con las órdenes **adelante** y **atrás,** que mueven la tortuga en el plano.
También podemos hacer que la tortuga gire hacia su derecha o hacia su izquier-


da. La orden que cumple en el espacio una función similar a la de **derecha** e **izquierda** en el plano es **virar.** Digamos:

?**virar** 25

¿Hacia dónde ha doblado la tortuga? Es necesario hacer aparecer la triada para poder percibir el giro de la tortuga. Escribamos:

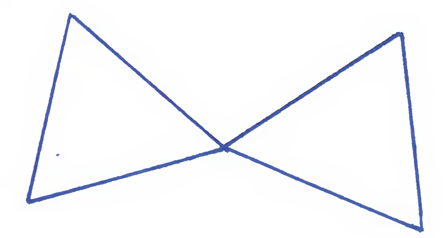
triada

¿Qué ha pasado con los ejes de la tortuga? El eje que antes era vertical, ha girado 25 grados hacia la izquierda. Lo mismo ha sucedido con el otro eje. El eje que apunta hacia afuera de la pantalla, ha girado sobre sí mismo, por lo tanto, permanece igual y en consecuencia, aún no lo podemos ver.
Usemos estas dos órdenes y hagamos, por ejemplo, un triángulo:

```
para triángulo
repetir 3[andar 50 virar 120]
fin
```

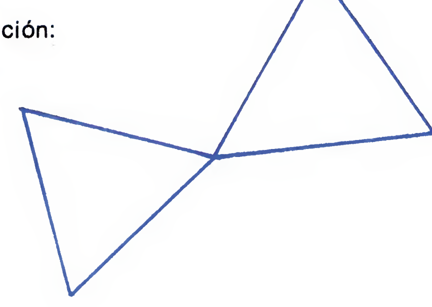
Dibujemos ahora una mariposa:

```
para mariposa
virar 50
triángulo
virar —160
triángulo
virar 110
fin
```



Probemos ahora ubicarla en otra posición:

```
? tri
? sp virar 35 andar —20 cp
? mariposa
```

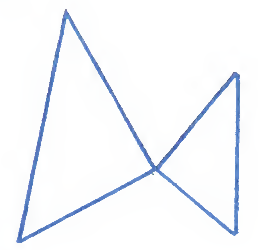


El dibujo nos es familiar: la mariposa ha girado 35 grados hacia la izquierda en su propio plano.

La tortuga sale al espacio

Las órdenes aprendidas —aunque mueven la tortuga por el espacio— producen hasta ahora efectos similares a los de la tortuga dibujando en el plano. Saquemos ahora la tortuga a dar un paseo por el espacio. Escribamos:

```
?tri
?rolar 60
?mariposa
```



¿Qué le ha pasado a la mariposa? Para poder apreciar el efecto de la orden **rolar,** imaginemos que la tortuga es un barco que se desplaza a través del océano. La orden **rolar 60** hace que el barco-tortuga escore **60** grados a estribor.

Nota: En náutica, se usan los términos babor y estribor para designar los costados izquierdo y derecho del barco, mirando de popa a proa. La proa es la parte delantera; la popa, la trasera. El término escorar indica la inclinación lateral de una embarcación.

Hagamos que la mariposa escore hacia un lado y hacia otro, simulando el balanceo de un barco:

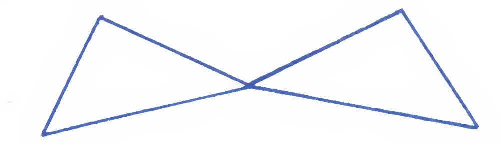
```
para balanceo
tri
rolar 60
esperar 50
mariposa
tri
rolar —60
esperar 50
mariposa
balanceo
fin
```

Nota: Para detener este procedimiento recursivo, use **CTRL · STOP**.

Siguiendo con la comparación anterior, el barco-tortuga no solo puede inclinarse de babor a estribor, sino también de proa a popa y viceversa. La orden que hace que la tortuga se incline hacia adelante o hacia atrás es **cabecear**. Probemos:




```
?tri
?cabecear 50
?mariposa
```



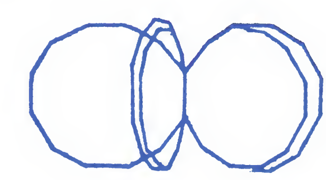
¿Percibe la diferencia entre esta figura y la anterior? Nuevamente la mariposa ha salido del plano de la pantalla. La parte superior de la mariposa se ha inclinado hacia atrás de la pantalla; la parte inferior, hacia adelante.
Estas dos nuevas órdenes, **rolar** y **cabecear**, sacan la tortuga del plano de la pantalla, si la tortuga se halla en él.

A distintas ubicaciones, imágenes diferentes

A partir de las órdenes **rolar** y **cabecear** podemos empezar a crear figuras tridimensionales. Tratemos de dibujar los pétalos de la flor del principio del capítulo. Escribamos:

```
para aro
repetir 12 [andar 8 virar 30]
fin

para pétalos
repetir 5 [aro rolar 72]
fin
```



¿Por qué los pétalos no aparecen exactamente a los de la figura del comienzo del capítulo? Imaginemos que la figura generada en la pantalla es la fotografía de unos pétalos situados enfrente y a la misma altura de la cámara fotográfica; los pétalos de adelante tapan a los de atrás y no podemos percibirlos en su totalidad. ¿Dónde tendrían que estar los pétalos para que nuestra cámara tomara la figura del comienzo del capítulo? Probemos dibujar primero el tallo:

```
para tallo
andar 40
fin
```

para flor
tallo
pétalos
fin
?flor

Ya tenemos nuestra flor. ¿Qué ha pasado? Como la tortuga ha avanzado 40 pasos antes de dibujar los pétalos, éstos se han ubicado en la parte superior de la pantalla y nuestra cámara los ha fotografiado desde abajo, logrando así una imagen idéntica a la inicial.
A distintas ubicaciones de la tortuga en la pantalla van a corresponder imágenes distintas de un mismo objeto.

Representación del dibujo tridimensional en la pantalla

En su recorrido por el espacio —al igual que en su recorrido por el plano— la tortuga deja un rastro o estela. La sucesión de puntos del itinerario de la tortuga genera la figura tridimensional buscada.

Para representar paso a paso esa sucesión de puntos en la pantalla bidimensional, el método adoptado es el de la proyección central o cónica que consiste, de manera resumida, en lo siguiente:

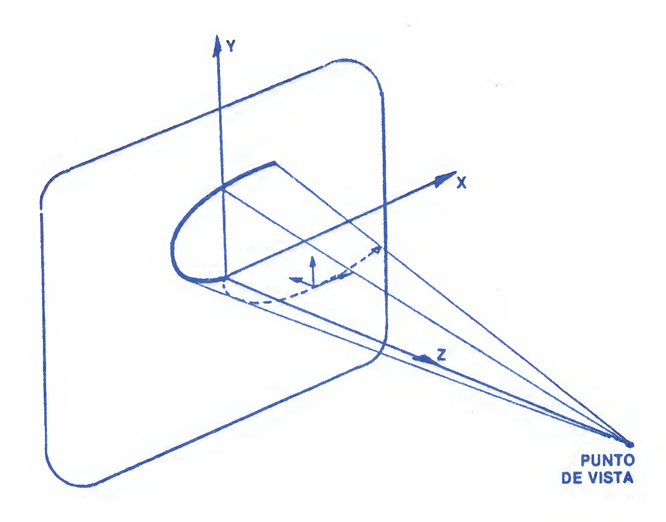
Supongamos un punto situado a una distancia determinada frente al centro de la pantalla. Unamos con rectas imaginarias ese punto con cada punto del recorrido de la tortuga y marquemos las intersecciones de esas rectas con la pantalla. El conjunto de puntos así generados es una imagen en dos dimensiones representativa de la figura trazada en el espacio. (Notemos que a cada punto del recorrido de la tortuga le corresponde un punto en la pantalla). El punto fijo situado frente a la pantalla se denomina punto de vista o centro de la proyección, y el haz de rectas emergentes de ese punto se denomina haz de rayos visuales ó proyectantes. La imagen generada es similar a la que se obtendría con una cámara fotográfica ubicada en ese punto.

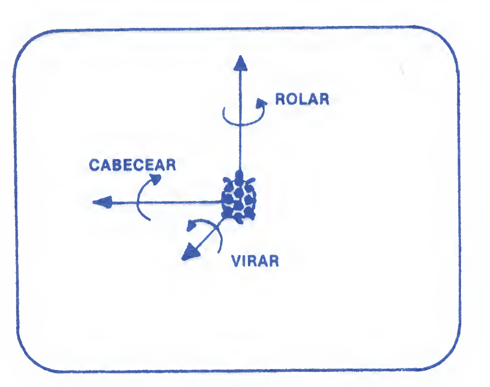
La orden **tri** prepara la escena para la realización de dibujos tridimensionales: borra la pantalla, prepara la tortuga para dibujar en el espacio, en estado visible y en el modo ventana y la ubica mirando hacia arriba.

La proyección cónica que genera la imagen tiene el punto de vista o centro de proyección a una distancia F = 360 del centro de la pantalla, medida en unidades de movimiento de traslación de la tortuga. Este valor es impuesto por la orden **tri** y puede ser modificado mediante la orden **hacer.** Su magnitud —al igual que la distancia focal de la lente de una cámara fotográfica— determina la apertura de los rayos que generan la imagen. Un valor similar al de la diagonal de la pantalla genera imágenes similares a las que se obtendrían con una cámara fotográfica con lente normal. Con un valor menor, las imágenes resultantes serían similares a las tomadas con gran angular; con un valor mayor, las imágenes se parecerían a las tomadas con teleobjetivo.

Al comenzar, la tortuga está en el centro y en el plano de la pantalla. Las coordenadas x, y, z de la tortuga están referidas a un sistema de ejes coordenados fijos, ubicado en el centro de la pantalla. Los ejes x horizontal, e y vertical, se hallan en el plano de la pantalla; el eje z es perpendicular a la pantalla y avanza hacia afuera (terna derecha).

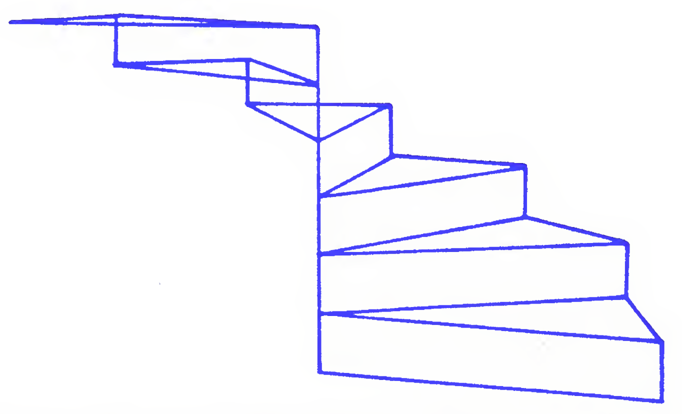
En la **triada** que representa la tortuga en su condición inicial, el eje longitudinal coincide con y, el transversal con x negativo (– x), y el restante coincide con z.





Cómo crear una escalera helicoidal

A primera vista, construir una escalera helicoidal como la de la figura siguiente parece un objetivo algo ambicioso. Sin embargo, utilizando la técnica de dividir el dibujo en partes pequeñas y manejables, podremos hacerla sin muchos problemas.



Para construir una escalera necesitamos escalones; por ello, nuestro primer paso será construir un escalón.
El escalón, a su vez, está formado por una parte vertical, la "contrahuella", y por una parte horizontal, la "huella" (el lugar en donde apoyamos el pie).
Para dibujar la contrahuella nos bastará un viejo y conocido rectángulo plano:

```
para contrahuella
repetir 2 [andar 15 virar – 90 andar 80 virar – 90]
fin
```



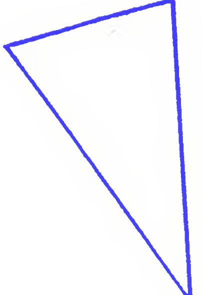
La huella en la escalera helicoidal tiene la forma de un triángulo; el vértice más agudo del triángulo se ubica junto al eje de la escalera.

```
para huella
andar 80
virar 105
andar 41,5
virar 105
andar 80
virar 150
fin
```



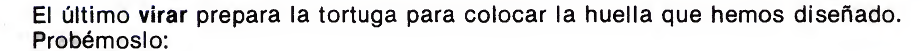
Probemos el procedimiento huella:

```
?tri sp anda – 20 cp
?huella
```



Tratemos ahora de armar un escalón, combinando contrahuella y huella. Para ello, luego de realizar la contrahuella, debemos ubicar la tortuga en su extremo superior y luego hacerla cabecear 90, de manera que coloque la huella en un plano perpendicular a la contrahuella. Realicemos para ello el procedimiento subir:

```
para subir
andar 15
cabecear 90
virar – 90
fin
```



El último **virar** prepara la tortuga para colocar la huella que hemos diseñado. Probémoslo:

```
?contrahuella
?subir
?huella
```

Al terminar la huella, la tortuga queda en la misma posición en que la dejó subir; es decir, en el plano de la huella. Orientémosla nuevamente según el eje de la escalera:

```
para volver.a.ubicar
virar 90
cabecear – 90
fin
```

Podemos ya juntar las partes y construir el escalón:

```
para escalón
contrahuella
subir
huella
volver.a.ubicar
fin
```

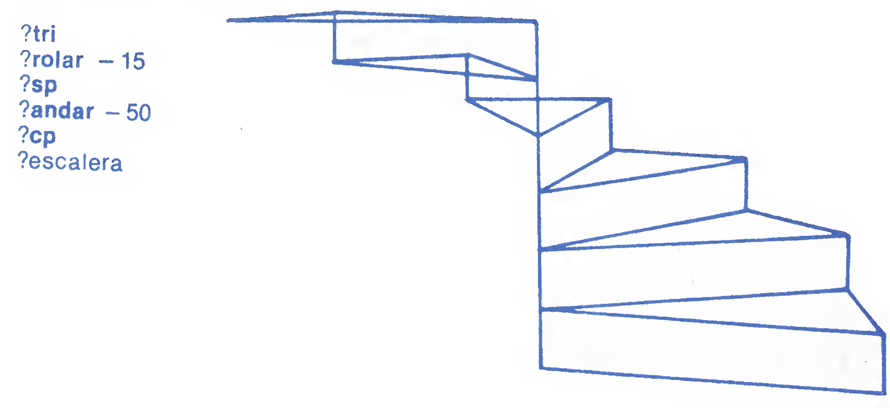
¿Es suficiente repetir varias veces el escalón para realizar la escalera?

Pruébelo.
¿Qué sucede? Los escalones se han apilado unos sobre otros. Para evitar que los escalones se superpongan, la tortuga debe rolar entre escalón y escalón:

```
para escalera
repetir 6 [escalón rolar 30]
fin
```

El último paso consistirá en ubicar la tortuga en el extremo inferior de la pantalla, haciéndola rolar un poco antes de que empiece a dibujar, a fin de mirar la escalera desde arriba y poder percibir mejor, de esa manera, la organización de los escalones. Probemos:

```
?tri
?rolar – 15
?sp
?andar – 50
?cp
?escalera
```



Desdoblamiento de las órdenes

Si se desea, se pueden utilizar, en lugar de las órdenes de movimiento ya vistas, un conjunto de órdenes abreviadas que incluyen en su nombre la indicación del sentido del movimiento. Es decir, se puede desdoblar cada una de las órdenes conocidas en dos nuevas órdenes que se corresponden con los dos sentidos de movimiento posibles. De esta manera, no es necesario utilizar argumentos negativos.*
Así, por ejemplo, el movimiento de traslación se puede expresar con **aad** (andar adelante) y **aat** (andar atrás). **aad** mueve la tortuga en el mismo sentido que **andar**; **aat** la mueve en sentido contrario.

** **El uso artesanal de las computadoras**, Horacio C. Reggini, Buenos Aires, Galápago, Dist. Emecé (en publicación).*

Las órdenes abreviadas elegidas son:

aad	(andar **ad**elante)	
aat	(andar **at**rás)	(Equivale a **andar** con argumento negativo)
vde	(virar **de**recha)	
viz	(virar **iz**quierda)	Equivale a **virar** con argumento negativo)
rde	(rolar **de**recha)	
riz	(rolar **iz**quierda)	(Equivale a **rolar** con argumento negativo)
cad	(cabecear **ad**elante)	
cat	(cabecear **at**rás)	(Equivale a **cabecear** con argumento negativo)

Ejemplo:

Podemos redefinir nuestra mariposa de la siguiente manera:

```
para triángulo
repetir 3 [aad 50 viz 120]
fin

para mariposa
viz 50
triángulo
vde 160
triángulo
viz 110
fin
```

Probemos:

```
?tri
?sp viz 35 aat 20 cp
?mariposa
```

Utilicemos las órdenes abreviadas para dibujar ahora una banqueta:

```
para banqueta
cad 90
repetir 4 [aad 20 pata vde 90]
fin

para pata
viz 45 cad 80
aad 40 aat 40
cat 80 vde 45
fin
```

Probemos:

```
?tri
?riz 20 cat 10
?banqueta
```



CAPÍTULO 7

LOGO Y LA MÚSICA

MSX-Logo brinda la posibilidad de escribir melodías y escucharlas, ofreciendo una nueva manera de acceder al mundo de la música. Para lograrlo, basta enseñar a la máquina las notas que debe ejecutar, de la misma manera que enseñamos a la tortuga a realizar figuras geométricas. Lo hacemos usando la orden **sonido**. Por ejemplo:

```
?sonido 0 440 10 30
```

genera un sonido para el canal 0, de frecuencia 440 (la), volumen 10 y duración 30. Es decir, sonido necesita 4 argumentos, que son:

Número de canal

MSX-Logo permite crear obras musicales a tres voces que suenen simultáneamente. Así podemos armonizar melodías y probarlas utilizando a MSX-Logo como si se tratara de un instrumento musical. El número de canal indica a Logo qué voz estamos escribiendo. Podemos ejecutar los tres canales en forma simultánea.
Ejemplo:

```
para acorde
sonido 0 262 10 50
sonido 1 330 10 50
sonido 2 392 10 50
fin
```

Hemos creado un acorde de do mayor. Ahora inventemos un compás:

```
para compás
sonido 0 392 15 40
sonido 1 330 15 40
sonido 2 262 15 60
sonido 0 349 15 40
sonido 1 294 15 40
sonido 2 247 15 20
sonido 0 392 15 40
```

```
sonido 1 330 15 40
sonido 2 262 15 40
fin
```

Nótese que hemos ido alternando los tres canales para lograr el efecto deseado: las notas de un acorde se deben escribir una junto a la otra para que suenen simultáneamente.

Número de frecuencia

Con este argumento le damos al MSX-Logo la altura del sonido que deseamos escuchar. El rango de frecuencias posibles abarca 7 octavas: desde 28 hasta 4095, lo que equivale al teclado completo de un piano (Ver la Tabla de frecuencias). Pero además es posible crear efectos especiales, como, por ejemplo, una sirena que es imposible lograr en un piano.
Ejemplo:

```
para sirena :n
si :n = 440 [parar]
sonido 0 :n 10 5
sirena :n + 1
fin

?sirena 220
```

Número de amplitud

Es un número que determina el volumen de sonido a emitir. Su rango va de 0 a 15. Un volumen 0 no emitirá sonido y puede utilizarse para crear silencios.
Ejemplo:

```
para probar.volumen :n
si :n > 15 [parar]
sonido 0 440 :n 30
probar.volumen :n + 1
fin

?probar. volumen 1
```

Número de duración

Es un número entero entre 0 y 255 que indica la duración que tiene el sonido a emitir. Si por ejemplo a una negra le damos una duración de 24 una corchea tendrá duración 12, una blanca 48, etc.

Tabla de frecuencias

La siguiente tabla muestra las frecuencias (redondeadas a enteros) de cuatro octavas de la escala temperada (un semitono entre notas). Una tabla más completa puede verse en el Anexo A.



Frecuencia	Nota
110	LA
117	LA #, SIb
123	SI
131	DO (DO bajo)
139	DO #,REb
147	RE
156	RE #,MIb
165	MI
175	FA
185	FA #,SOLb
196	SOL
208	SOL #,LAb
220	LA (debajo DO central)
233	LA #,SIb
247	SI
267	DO (DO central)
277	DO #,REb
294	RE
311	RE #,MIb
330	MI
349	FA
370	FA #,SOLb
392	SOL
415	SOL #,LAb

Frecuencia	Nota
440	LA (sobre DO central)
466	LA #,SIb
494	SI
523	DO (DO alto)
554	DO #,REb
587	RE
622	RE #,MIb
659	MI
698	FA
740	FA #,SOLb
784	SOL
831	SOL #,LAb
880	LA (sobre DO alto)
932	LA #,SIb
988	SI
1047	DO
1109	DO #,REb
1175	RE
1245	RE #,MIb
1319	MI
1397	FA
1480	FA #,SOLb
1568	SOL
1661	SOL #,LAb
1760	LA

Un ejemplo

El ejemplo utilizado es una versión simplificada del **Minué en sol mayor** de J.S. Bach. Sólo se ha escrito la primera parte; dejamos al lector la tarea de completarlo. Se han definido previamente las notas a utilizar con breves procedimientos en los que se fijan la frecuencia y la amplitud y se dan como argumentos el número de canal y la duración de la corchea. Diferenciamos las distintas octavas agregando al nombre de la nota una a (alto) si está sobre el do central, una c (central) si está en la parte central y una b (bajo) si está por debajo. Nuestra escala va del lab (la bajo) hasta el sola (sol alto), es decir tiene un registro de 3 octavas. Como el **Minué** está en sol mayor hemos definido también el fa #.
Ejemplo:

```
para lab :c :d
sonido :c 110 10 :d
fin

para fac# :c :d
sonido :c 370 10 :d
fin

para lac :c :d
sonido :c 220 10 :d
fin

para sola :c :d
sonido :c 784 10 :d
fin
```

Para hacer más comprensible el procedimiento lo hemos dividido en 5 partes y en cada una hemos escrito dos compases.

```
para parte 1 :d
rea   0 :d * 2  solb  1 :d * 4
solc  0 :d      laa   0 :d
sia   0 :d      lac   1 :d * 2  doa 0 :d
rea   0 :d * 2  sic   1 :d * 6
solc  0 :d * 2  solc  1 :d * 2
fin

para parte 2 :d
mia   0 :d * 2  doc   1 :d * 6
doa   0 :d      rea   0 :d
mia   0 :d      faa # 0 :d
sola  0 :d * 2  sic   1 :d * 6
solc  0 :d * 2  solc  0 :d * 2
fin

para parte 3 :d
doa   0 :d * 2  lac   1 :d * 6
rea   0 :d      doa   0 :d
sia   0 :d      laa   0 :d
sia   0 :d * 2  solb  1 :d * 6
doa   0 :d      sia   0 :d  6
laa   0 :d      solc  0 :d
fin
```

Como estos 6 compases se van a repetir, los agrupamos en un procedimiento:

```
para frase.repetida :d
parte1 :d
parte2 :d
parte3 :d
fin
```

Ahora escribimos lo que falta:

```
para parte4 :d
fac # 0 :d * 2  rec   1 :d * 2
solc  0 :d      sic   1 :d * 2  laa   0 :d
sia   0 :d      solb  1 :d * 2  solc  0 :d
sia   0 :d * 2  rec   1 :d * 2
laa   0 :d * 4  reb   1 :d      doc   1 :d
sic   1 :d      lac   1 :d
fin

para parte5 :d
laa   0 :d * 2  doc   1 :d * 2
sia   0 :d      rec   1 :d * 2  laa   0 :d
```



```
solc  0 :d      reb   1 :d * 2  fac # 0 :d
solc  0 :d * 6  solb  1 :d * 6
fin
```

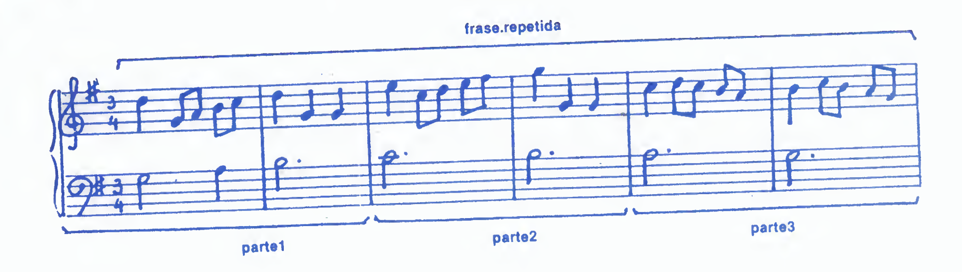
Escribamos ahora el procedimiento principal al que llamaremos Minué:
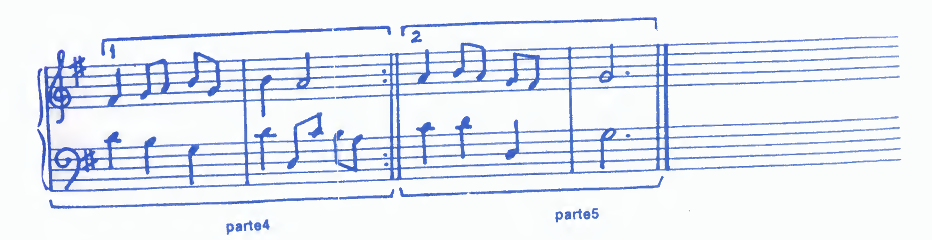
```
para Minué :d
frase.repetida :d
parte4 :d
frase.repetida :d
parte 5 :d
fin
```

Si queremos escucharlo, escribimos:

```
?Minué 12
```

Y escucharemos nuestra bella obra. El número 12 indica la duración de la corchea. Si queremos que suene más lento, debemos darle un número mayor como argumento.
La partitura correspondiente a nuestro ejemplo es una adaptación a dos voces de la primera parte del **Minué en Sol Mayor** de Juan Sebastián Bach del libro de Ana Magdalena Bach. Para simplificar, se ha repetido la primera parte cambiando sólo el final; en el original, el bajo cambia en la repetición.







CAPÍTULO 8

DESCRIPCIÓN DE LAS PRIMITIVAS

Este capítulo comprende las siguientes secciones:
- Descripción de los argumentos
- Descripción de las primitivas
- Operaciones
- Caracteres especiales
- Primitivas especiales

Las primitivas MSX-Logo se han ordenado alfabéticamente. En la explicación se da, junto al nombre de la primitiva, su abreviatura (si corresponde) y se señala también el tipo de argumento que utiliza, de acuerdo a la tabla de la página siguiente. Al final de cada explicación se incluye un breve ejemplo de uso.
Las primitivas se han clasificado en órdenes de comando y órdenes de operación. Denominamos órdenes de operación u operaciones a aquéllas que responden con algo (que puede ser, por ejemplo, utilizado como argumento de otra orden). Así **suma** responde con un número que es la suma de los números datos. Las órdenes de comando o comandos no responden con algo sino que producen por sí solas una acción. Por ejemplo, **escribir**, hace aparecer su argumento en la pantalla.
En toda línea Logo, la primera palabra debe ser un comando: una operación sólo puede escribirse como argumento de otra orden. Por ejemplo:

```
?esc suma 3 7
10
```

Descripción de los argumentos

Tipo de argumento	Significado
acontecimiento	un número entero entre 0 y 5.
amplitud	un número entero entre 0 y 15.
archivo	nombre de un archivo de disco o casete. La palabra no debe tener más de 8 caracteres en el primer caso, o de 6 en el segundo, y no puede incluir puntos ni comas.
byte	un número entero entre 0 y 255.
canal	un número entero de 0 a 2.

carácter	un símbolo ortográfico.
código de figura	lista formada por 32 números enteros.
columna	un número entero entre 0 y 28.
dirección	un número entero entre 0 y 65535.
duración	un número entero entre 0 y 255.
frecuencia	un número entero entre 28 y 4095.
línea	un número entero entre 0 y 23.
lista	información encerrada entre dos corchetes ([ ]).
lista de instrucciones	lista formada por procedimientos Logo.
lista de nombres	lista formada con nombres de procedimientos, variables o lista de propiedades.
lista de tortugas	lista formada por números enteros comprendidos entre 0 y 29.
nombre	palabra que identifica a un procedimiento, variable o lista de propiedades.
número de color	un número entero entre 0 y 15.
número de figura	un número entero entre 0 y 59.
número de mando	un número entero (1 ó 2).
número de toma	un número entero entre 0 y 255.
número de tortuga	un número entero entre 0 y 255.
objeto	una palabra, una lista o un número.
palabra	una palabra (secuencia de caracteres sin espacios intermedios).
predicado	objeto que sólo puede ser **cierto** o **falso**.
propiedad	propiedad.
tono	un número entero entre 0 y 31.
x	número representativo de la coordenada x (entre – 128 y 127).
y	número representativo de la coordenada y (entre – 69 y 70).

Descripción de las primitivas

activar (número de tortuga) comando
activar (lista de tortugas)

Indica a Logo cuál es (o cuáles son) la o las tortugas que deben cumplir las órdenes que se den de aquí en adelante.
Ejemplo:

?**activar** 10
?**ffig** 2 **fcolor** 6 **mt ad** 30 **de** 90
?**activar** 11
?**ad** 30 **de** 90 **ad** 20 **ffig** 2 **fcolor** 10 **mt**
?**activar** [10 11]
fvel 5

adelante, ad (número) comando
Mueve hacia adelante la o las tortugas activas, haciéndoles avanzar los pasos de tortuga indicados.

Ejemplo:

?**adelante** 20

al.contacto (número de tortuga) (número de tortuga) (lista de instrucciones) comando
Establece un demonio que vigila si se produce un choque entre las dos tortugas indicadas. Cuando esto sucede, el demonio interrumpe el programa en ejecución y cumple la lista de instrucciones dato. Luego, se reanuda la ejecución dle programa interrumpido.
Ejemplo:

?**al.contacto** 11 12 [congelar esc [el perro atrapó al gato]]

arctan (número) operación
Responde con el ángulo cuya tangente es el número dato. El resultado está comprendido entre – 90 y 90.
Ejemplo:

?**esc arctan** 1
45

ascii(carácter) operación
Responde con el número de código que corresponde al carácter de acuerdo al Código Nacional Americano de Normas para el Intercambio de Información (en inglés, **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange, **ASCII**). Ver la Tabla de caracteres en el Anexo A.
Ejemplo:

?**esc ascii** "a
97

atrás,at(número) comando
Mueve hacia atrás la o las tortugas activas, haciéndolas retroceder los pasos de tortuga indicados.
Ejemplo:
?**atrás** 50

azar(número) operación
Responde con un número entero seleccionado al azar, comprendido entre 0 y el número dado como argumento menos 1.
Ejemplo:
?**ad azar** 30

bg comando
Significa **b**orrar **g**ráficos. Borra lo que está dibujado en la pantalla y lleva la o las tortugas activas del centro.

boarchivo (nombre de archivo) comando
Borra del disco el archivo especificado por el argumento.
Ejemplo:

?**boarchivo** "personal

bods comando
Significa **bo**rrar **d**emonio**s**. Borra todos los demonios creados por las órdenes **al.contacto** y/o **cuando**.

bon (nombre de variable) comando
bon (lista de nombres de variables)
Significa **bo**rrar **n**ombre. Borra del espacio de trabajo la o las variables especificadas como argumento.
Ejemplo:

?**bon** "lado
?**bon** [radio ángulo]

bons comando
Significa **bo**rrar **n**ombre**s**. Borra todas las variables del espacio de trabajo.

boprops comando
Significa **bo**rrar **prop**iedade**s**. Borra todas las listas de propiedades del espacio de trabajo.

bops comando
Significa **bo**rrar **p**rocedimiento**s**. Borra todos los procedimientos del espacio del trabajo.

borrar,bo(nombre de procedimiento) comando
borrar,bo(lista de nombres de procedimientos)
Borra del espacio de trabajo el o los procedimientos especificados como argumento.
Ejemplo:

?**borrar**"cuadrado
?**bo** [casa árbol nube]

botodo comando
Significa **bo**rrar **todo**. Borra todos los procedimientos, variables y listas de propiedades del espacio de trabajo. Equivale a **bops bons boprops**.

botón? (número de mando) operación



Responde **cierto** si se ha apretado el botón del mando especificado. En caso contrario responde **falso**.
Ejemplo:

?**si botón?** 2 [pedir 5 [ad 50]]

bp comando
Significa **b**orrar **p**antalla. Borra los textos y dibujos de la pantalla, lleva la o las tortugas activas al centro, les pone velocidad 0 y las deja apuntando hacia arriba (rumbo 0). **bp** equivale a **limpiar sp fvel** 0 **fpos** [0 0] **frumbo** 0 **bt cp.**

bt comando
Significa **b**orrar **t**exto. Borra el texto de la pantalla.

cada (lista de instrucciones) comando
Hace que cada una de las tortugas ejecute secuencialmente la lista de instrucciones especificada.
Ejemplo:

?**activar** [0 1 2 3 4]
?**sp ffig** 3 **mt**
?**cada** [**fy azar** 80]

cambiar.color (número de color) (número de color) comando
Todos los gráficos y textos de la pantalla que tienen el color indicado por el primer argumento, cambian al color indicado por el segundo.
Ejemplo:

?**cambiar.color** 15 1

caracter, car (número) operación
Responde con el caracter cuyo código ASCII se especifica como argumento. Ver la Tabla de caracteres en el Anexo A.
Ejemplo:

?**esc carácter** 99
c

centro comando
Detiene la o las tortugas activas y las lleva al centro de la pantalla, a la posición [0 0] y rumbo 0.
Ejemplo:

?**pedir** 5 **centro**

clerto operación
Es un nombre predefinido por Logo. Es el valor que se asigna a las frases lógicas cuando son ciertas.
Ejemplo:

```
?hacer "prueba cierto
si :prueba = cierto [ad 100]
```

cociente (número) (número) operación
Responde con el resultado de dividir el primer argumento por el segundo.
Ejemplo:

```
?esc cociente 875 5
175
```

color operación
Responde con el número del color de la tortuga activa.
Ejemplo:

```
?esc color
15
```

color.debajo operación
Responde con el número del color existente debajo de la pluma de la tortuga activa.
Ejemplo:

```
?esc color.debajo
5
```

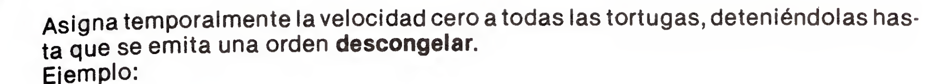
colorf operación
Significa **color** fondo. Responde con el número del color del fondo.
Ejemplo:

```
?esc colorf
1
```

colorp operación
Significa **color** pluma. Responde con el número del color que tiene la pluma en ese momento.
Ejemplo:

```
?esc colorp
9
```

congelar comando


Asigna temporalmente la velocidad cero a todas las tortugas, deteniéndolas hasta que se emita una orden **descongelar**.
Ejemplo:

```
?activar 2 mt fvel 5
?congelar
```

conimpresora comando
Activa la impresora.
Ejemplo:

```
?conimpresora
?esc [que tal]
```

conpluma, cp comando
Hace que la o las tortugas activas dibujen al desplazarse.
Ejemplo:

```
?cp ad 30
```

convertir comando
Sustituye la figura de la o las tortugas activas por el dibujo de la pantalla que se encuentra debajo de ellas.
Ejemplo:

```
?iz 45 repetir 4 [ad 3 de 90]
?convertir                    (la tortuga no se ve)
?ad 50                        (la tortuga ha tomado la figura de un
                              rombo)
```

coorx operación
Significa **coor**denada **x**. Responde con la coordenada x de la posición de la o las tortugas activas.
Ejemplo:

```
?iz 30 ad 50
?esc coorx
-25
```

coory operación
Significa **coor**denada **y**. Responde con la coordenada y de la posición de la o las tortugas activas.
Ejemplo:

```
?iz 30 ad 50
```





?**esc coory**
43,30127

---

copiardef(nombre de procedimiento)(nombre) comando
Significa **copiar def**inición. Crea un nuevo procedimiento cuyo nombre es el segundo argumento y su definición es copia del procedimiento nombrado en primer término.
Ejemplo:

?**copiardef** "cuadrado "c

---

copiarfig (número de figura) (número de figura) comando
Significa **copiar fig**ura. Copia la primera figura en la segunda.
Ejemplo:.

?**copiarfig** 3 15

---

cos (número) operación
Responde con el coseno del ángulo dado como argumento.
Ejemplo:

?**esc cos** 45
0,70710678

---

cosa (nombre de variable) operación
Devuelve el contenido de la variable especificada.
Ejemplo:

?**hacer** "apellido "Rodríguez
?**esc cosa** "apellido
Rodríguez

---

cuando (acontecimiento) (lista de instrucciones) comando
Activa un demonio que, si ocurre el acontecimiento especificado, ejecuta la lista de instrucciones. Consultar la tabla de acontecimientos que aparece en la sección Demonios del Capítulo 4, Las tortugas móviles o el Anexo A.
Ejemplo:

?**cuando** 0 [**ad** 2 **de azar** 10]

---

cuenta (objeto) operación
Responde con la cantidad de elementos del objeto, que puede ser una palabra, una lista o un número.
Ejemplo:


?**esc cuenta** [Alfredo Juan 12]
3
?**esc cuenta** "esteesunnombremuylargo
22

---

cumplir (lista) comando u operación
Ejecuta la lista. Si la lista es una operación, cumplir devuelve el resultado.
Ejemplo:

?**esc cumplir** [**sen** 30 * **cos** 60]
0,25881904
?**cumplir** [**ad** 20 **iz** 90]

---

cursor operación
Responde con la posición del cursor como lista de dos números (la columna y la línea).
Ejemplo:

?**bt esc cursor**
0 0

---

definido? (nombre de procedimiento) operación
Responde **cierto** si el argumento es un procedimiento definido. En caso contrario responde **falso**.
Ejemplo:

?**si definido**? "cuadrado cumplir [**ed** "cubo]
para cubo
fin

---

definir (nombre) (lista de instrucciones) comando
Define un procedimiento sin usar el Editor; es utilizado generalmente dentro de otro procedimiento. El primer argumento es el nombre que llevará el procedimiento y el segundo es la lista de las instrucciones. Si el procedimiento tuviera alguna variable, el nombre de ésta deberá ser el primer elemento de la lista de instrucciones; si no hubiera variables, el primer elemento deberá ser una lista vacía.

?**definir** "mensaje [[nombre] [**esc** :nombre] [**esc** "ganaste!!!]]
?mensaje "Jorge
Jorge
ganaste !!!

---

derecha.de (número) comando
Gira la o las tortugas activas hacia la derecha el número de grados indicado por su argumento.

Ejemplo:

?**derecha** 90

descongelar comando
Activa las tortugas detenidas, devolviéndoles la velocidad que tenían antes de la orden **congelar**.
Ejemplo:

?**fvel** 10 **esperar** 200 **congelar**
?**descongelar**

dif (número) (número) operación
Responde con el resultado de restar el segundo argumento al primero.
Ejemplo:

?**esc dif** 23456,7 12345,6
11111,1

dir comando
Significa **dir**ectorio. Lista los nombres de todos los archivos contenidos en el disco.

distancia [x y] operación
Responde con la distancia, medida en pasos de tortuga, entre la posición actual de la tortuga activa y el punto indicado por el argumento.
Ejemplo:

?**centro esc distancia** [50 50]
70,710678

edfig (número de figura) comando
Significa **ed**itar **fig**ura. Ingresa en el Editor de figuras, donde coloca la definición de la figura indicada.
Ejemplo:

?**edfig** 5

editar, ed (nombre de procedimiento) comando
editar, ed (lista de nombres de procedimientos)
Ingresa en el Editor, donde coloca las definiciones de los procedimientos indicados para su edición.
Ejemplo:

?**editar** [cuadrado rombo]

para cuadrado
fin

para rombo
fin

edns comando
Significa **ed**itar **n**ombre**s**. Ingresa en el Editor, donde coloca los nombres de todas las variables y sus respectivos valores.
Ejemplo:

?**edns**
hacer "perro 3
hacer "gato 2

ent (número) operación
Significa **ent**ero. Responde con la parte entera del número dado, truncando la parte decimal.
Ejemplo:

?**esc ent** 456,89
456

escribir, esc (objeto) comando
(**escribir** (objeto) (objeto) ...)
(**esc** (objeto) (objeto) ...)
Escribe en la pantalla sus argumentos y lleva el cursor al principio de la línea siguiente.
Ejemplo:

?**esc** [algarrobo cedro tilo]
algarrobo cedro tilo
?(**esc** [algarrobo cedro tilo] [son árboles])
algarrobo cedro tilo son árboles

escribirs (objeto) comando
(**escribirs** (objeto) (objeto) ...)
Significa **escribir s**eguido. Imprime sus argumentos sin pasar a la línea siguiente. Si son listas, no imprime los corchetes extremos.
Ejemplo:

?**escribirs** "azucena
azucena?
?(**escribirs** "azucena "rosa "lirio)
azucenarosalirio?

espacio operación

Responde la cantidad de memoria disponible. El primer valor es el espacio disponible para nombres y variables, y el segundo el correspondiente a procedimientos.

Ejemplo:

```
?esc espacio
5759 12268
```

esperar (número) comando

Provoca una pausa de duración número / 50 segundos.

Ejemplo:

```
?esperar 500 esc "hola
hola                    (el mensaje aparece 10 segundos más tarde)
```

estampar comando

Estampa en la pantalla una copia de la figura de la o las tortugas activas, que deben estar **conpluma**.

Ejemplo:

```
?de 90 ffig 3 fvel 10
?cp repetir 5 [esperar 50 estampar]
```

falso operación

Es un nombre predefinido por Logo. Es el valor que se asigna a las frases lógicas cuando son falsas.

Ejemplo:

```
?hacer "prueba falso
?si :prueba = falso [de 90]
```

fcolor (número de color) comando

Significa **f**ijar **color**. Colorea la figura de la o las tortugas activas con el color indicado por su argumento.

Ejemplo:

```
?fcolor 11
```

fcolorf (número de color) comando

Significa **f**ijar **color f**ondo. Colorea el fondo de la pantalla con el color indicado por su argumento.

Ejemplo:

```
?fcolorf 6
```



fcolorp (número de color) comando

Significa **f**ijar **color p**luma. Cambia el color de la pluma de la o las tortugas activas al indicado por su argumento.

Ejemplo:

```
?fcolorp 1
```

fcursor [columna línea] comando

Significa **f**ijar **cursor**. Ubica el cursor en el lugar de la pantalla establecido por su argumento.

Ejemplo:

```
?fcursor [10 10]
```

ffig (número de figura) comando

Significa **f**ijar **fig**ura. Hace que la o las tortugas activas adopten la figura especificada por su argumento.

Ejemplo:

```
?ffig 7
```

fig operación

Responde con el número de la figura de la o las tortugas activas.

Ejemplo:

```
?esc fig
3
```

fin comando

Es un nombre predefinido por Logo. Indica el final de la definición de un procedimiento.

fpos [x y] comando

Significa **f**ijar **pos**ición. Ubica la o las tortugas activas en la posición especificada por su argumento sin variar su orientación. Si la tortuga está **conpluma**, dejará un rastro al moverse.

Ejemplo:

```
?fpos [80 60]
```

fproporción (número) comando

Significa **f**ijar **proporción**. Establece una nueva longitud para los pasos verticales de la tortuga. Se utiliza en general para corregir distorsiones causadas por el monitor. (Ver **proporción**).

Ejemplo:

?**proporción** 0,8

---

frase, fr (objeto) (objeto) operación
(**frase** (objeto) (objeto) ...)
(**fr** (objeto) (objeto) ...)
Responde con una lista formada por la combinación de sus argumentos. Si los argumentos son a su vez listas, **frase** saca los corchetes extremos y forma una sola lista.
Ejemplo:

?**mostrar frase** "caballo [perro gato tortuga]
[caballo perro gato tortuga]
?**mostrar frase** [lápiz compás] [lapicera [goma de borrar] sacapuntas]
[lápiz compás lapicera [goma de borrar] sacapuntas]

---

frumbo (número) comando
Significa **f**ijar **rumbo**. Impone a las tortugas activas el rumbo especificado por su argumento. (Ver **rumbo**).
Ejemplo:

?**frumbo** 270 (apunta la tortuga hacia la izquierda)

---

ftexto (línea) comando
Significa **f**ijar **texto**. Borra la pantalla y hace que el texto que se escriba a continuación comience a partir de la línea indicada por su argumento. La línea superior de la pantalla es la 0 y la inferior la 23.
Ejemplo:

?**ftexto** 18

---

fvel (número) comando
Significa **f**ijar **vel**ocidad. Confiere a la o las tortugas activas la velocidad indicada por su argumento. La velocidad debe estar comprendida entre – 128 y 128.
Ejemplo:

?**fvel** 5

---

fvelx (número) comando
Significa **f**ijar **vel**ocidad **x**. Establece como componente x (horizontal) de la velocidad de la o las tortugas activas el valor indicado por su argumento. Como la componente y (vertical) de la velocidad no se modifica, el rumbo cambia.
Ejemplo:

?**frumbo** 45 **fvel** 10
?**fvelx** 0

---

fvely (número) comando
Significa **f**ijar **vel**ocidad **y**. Establece como componente y (vertical) de la velocidad de la o las tortugas activas el valor indicado por su argumento. Como la componente x (horizontal) de la velocidad no se modifica, el rumbo cambia.
Ejemplo:

?**frumbo** 45 **fvel** 10
?**fvely** 0

---

fx (x) comando
Significa **f**ijar coordenada **x**. Ubica la o las tortugas activas en la coordenada x especificada por su argumento; la coordenada y no se modifica.
Ejemplo:

?**fx** 30

---

fy (y) comando
Significa **f**ijar coordenada **y**. Ubica la o las tortugas activas en la coordenada y especificada por su argumento; la coordenada x no se modifica.
Ejemplo:

?**fy** 17

---

guardar (archivo) comando
Graba en el disco una copia del espacio de trabajo (procedimientos, variables y listas de propiedades) en un archivo con el nombre indicado por su argumento.
Ejemplo:

?**guardar** "polígono

---

guardarc (archivo) comando
Significa **guardar** en **c**asete. Graba en un casete una copia del espacio de trabajo (procedimientos, variables y listas de propiedades) en un archivo con el nombre indicado por su argumento. Es imprescindible que esté apretada la tecla de grabación del grabador.
Ejemplo:

?**guardac** "flores

---

guardardib (archivo) comando



Significa **guardar dib**ujo. Graba la imagen de la pantalla en un archivo de disco con el nombre indicado por su argumento.
Ejemplo:

?guardardib "ciudades

---

hacer (nombre) (objeto) — comando
Crea una variable que tiene como nombre el primer argumento y le asigna como valor el segundo argumento.
Ejemplo:

?hacer "Australia "canguro
?esc :Australia
canguro

---

hacia [x y] — operación
Responde con el rumbo que debería tener la tortuga para ir directamente desde la posición actual hasta la indicada por el argumento.
Ejemplo:

?centro esc hacia [50 50]
45,000654

---

igual? (objeto) (objeto) — operación
Responde **cierto** cuando los argumentos son iguales; en caso contrario responde **falso**.
Ejemplo:

?esc igual? colorf 5
cierto

---

imarchivo (archivo) — comando
Significa **imprimir archivo**. Imprime en la pantalla el archivo indicado por el argumento.
Ejemplo:

?imarchivo "polígono

---

imds — comando
Significa imprimir demonios. Imprime todas las órdenes de demonios vigentes.
Ejemplo:

?imds
al.contacto 1 2 [**de** 90]
al.contacto 5 6 [**fcolor** 10]

---

imns — comando
Significa **im**primir **n**ombre**s**. Imprime en la pantalla los nombres y valores de todas las variables.
Ejemplo:

?imns
hacer "frase [qué lindo día]
hacer "Australia "canguro

---

imp (nombre) — comando
imp (lista de nombres)
Significa **imp**rimir. Imprime en la pantalla las definiciones de los procedimientos indicados por su argumento.
Ejemplo:

?imp "cuadrado
?imp [cuadrado rombo]

---

improps — comando
Significa **im**primir **prop**iedades. Imprime en la pantalla todas las propiedades existentes en el espacio de trabajo.
Ejemplo:

?improps
pprop "manzana "color "rojo
pprop "manzana "aroma "suave
pprop "naranja "color "naranja
pprop "naranja "aroma "fuerte

---

imps — comando
Significa **im**primir **p**rocedimiento**s**. Imprime en la pantalla las definiciones de todos los procedimientos existentes en el espacio de trabajo.

---

imtodo — comando
Significa **im**primir **todo**. Imprime en la pantalla todos los procedimientos, variables y propiedades existentes en el espacio de trabajo.

---

imts — comando
Significa **im**primir **t**ítulo**s**. Imprime en la pantalla los nombres de todos los procedimientos existentes en el espacio de trabajo.

---

item (número) (objeto) — operación
Responde con el elemento del segundo argumento cuyo número de orden es el indicado por el primer argumento.

Ejemplo:

?**esc ítem** 3 [llama alpaca guanaco]
guanaco
?**esc ítem** 3 "guanaco
a

---

izquierda, iz (número) comando
Gira la o las tortugas activas hacia la izquierda el número de grados indicado por su argumento.
Ejemplo:
?**izquierda** 60

---

lc operación
Significa **l**eer **c**arácter. Espera que se pulse una tecla y devuelve el carácter que representa.
Ejemplo:

?**si lc** = "a [adelante 50]

---

limpiar comando
Borra los gráficos de la pantalla sin afectar la o las tortugas activas.

---

lista (objeto) (objeto) operación
(**lista** (objeto) (objeto) ...)
Responde con una lista formada por sus argumentos.
Ejemplo:

?**mostrar lista** [esta es] [una lista]
[[esta es] [una lista]]
?**mostrar** (**lista** "paloma "zorzal "canario "picaflor)
[paloma zorzal canario picaflor]

---

lista? (objeto) operación
Responde **cierto** cuando el objeto es una lista; en caso contrario responde **falso**.
Ejemplo:

?**esc lista?** [llavero encendedor pulsera]
cierto

---

listaprops (nombre) operación
Significa **lista** de **prop**iedades. Responde con la lista de propiedades asociada a su argumento; la lista de propiedades está compuesta por los nombres y sus correspondientes valores.

Ejemplo:

?**mostrar listaprops** "manzana
[color rojo sabor dulce aroma suave]

---

ll operación
Significa **leer línea**. Lee una línea del teclado y la responde como lista.
Ejemplo:

?**hacer** "frase **ll mostrar** :frase
qué lindo día
[qué lindo día]

---

mando (número de mando) operación
Responde con un número de 0 a 8, de acuerdo a la posición de la palanca del mando especificado.
Ejemplo:

?**esc mando** 1
4

---

mando? (número de mando) operación
Responde **cierto** cuando la palanca del mando especificado no se encuentra en el centro; en caso contrario responde **falso**.
Ejemplo:

?**si mando?** 1 [**frumbo mando** 1 * 45]

---

menosprimero, mp (objeto) operación
Responde el argumento sin su primer elemento.
Ejemplo:

?**esc menosprimero** "Paula
aula
?**mostrar** mp [bosque lago sierra]
[lago sierra]

---

menosúltimo, mu (objeto) operación
Responde el argumento sin su último elemento.
Ejemplo:

?**esc menosúltimo** "Paula
Paul
?**mostrar mu** [bosque lago sierra]
[bosque lago]

minus (número) operación
Responde su argumento cambiado de signo.
Ejemplo:

?**esc minus** 5
– 5

mostrar (objeto) comando
Imprime en la pantalla su argumento y lleva el cursor al principio de la línea siguiente. Es similar a **escribir**, pero no quita los corchetes extremos a las listas.
Ejemplo:

?**mostrar** [algarrobo cedro tilo]
[algarrobo cedro tilo]

mt comando
Significa **m**ostrar **t**ortuga. Hace visible la figura de la o las tortugas activas.

nivelsuperior comando
Detiene la ejecución de todos los procedimientos y devuelve el control al modo directo de Logo (también llamado "nivel superior"). Se diferencia de **parar**, en que este último sólo detiene la ejecución del procedimiento en que se encuentra.

no (predicado) operación
Responde **cierto** si su argumento es **falso**. Si es **cierto** responde **falso**.
Ejemplo:

?**esc no igual?** 4 5
cierto

nombrar (objeto) (nombre) comando
Crea una variable que tiene como nombre el segundo argumento y le asigna como valor el primero. Es equivalente a **hacer**, con los argumentos en distinto orden.
Ejemplo:

?**nombrar** "canguro "Australia
?**esc** :Australia
canguro

nombre? (nombre) operación
Responde **cierto** si el argumento es un número; en caso contrario responde **falso**.
Ejemplo:



?**esc nombre?** "apellido
cierto

número? (objeto) operación
Responde **cierto** si el argumento es un número en caso contrario responde **falso.**
Ejemplo:

?**esc número?** "casa
falso

o (predicado) (predicado) operación
(**o** (predicado) (predicado) ...)
Responde **cierto** si alguno de sus argumentos es cierto; si son todos falsos responde **falso**.
Ejemplo:

?**si o coory** > – 70 **coory** < 70 [**esc** "adentro]

obfig (número de figura) operación
Significa **ob**tener código de **fig**ura. Responde con una lista que representa en un cierto código a la figura indicada por su argumento. (Ver el Anexo C El código de las figuras).
Ejemplo:

?**esc obfig** 40
0 1 0 1 7 47 63 15 15 15 7 3 2 3 0 0 0 128 128 206 254 254 240 240 112 252 228 128 0 0 0 0

obprop (nombre) (propiedad) operación
Significa **ob**tener **prop**iedad. Responde con el valor de la propiedad indicado.
Ejemplo:

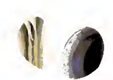
?**esc obprop** "manzana "sabor
dulce

ot comando
Significa **o**cultar **t**ortuga. Hace invisible la figura de la o las tortugas activas.

palabra (palabra) (palabra) operación
(**palabra** (palabra) (palabra) ...)
Responde con una sola palabra formada por la combinación de sus argumentos.
Ejemplo:

?**esc palabra** "saca "puntas
sacapuntas





?**esc** (**palabra** "corre "ve "y "dile)
correveydile

---

palabra? (objeto) — operación

Responde **cierto** si el argumento es una palabra. En caso contrario responde **falso**.
Ejemplo:

?**esc palabra?** "casa
cierto

---

para — comando
para (nombre)
Indica a Logo que se inicia la definición de un procedimiento.

---

parar — comando

Detiene la ejecución de un procedimiento devolviendo el control al procedimiento que lo llamó. Si no fue llamado por otro procedimiento, Logo retorna al "nivel superior". Sólo se utiliza dentro de procedimientos.

---

pb — comando

Significa **p**luma de **b**orrar. Hace la o las tortugas activas borren las líneas que cruzan al desplazarse. Para volver al estado normal se debe ordenar **conpluma**.
Ejemplo:

?**ad** 60
?**pb at** 60

---

pedir (número de tortugas) (lista de instrucciones) — comando
pedir (lista de tortugas) (lista de instrucciones)
Hace que las tortugas designadas por el primer argumento cumplan la lista de instrucciones. La lista de tortugas actualmente activas (las llamadas con **activar**) no se modifica.
Ejemplo:

?**pedir** 12 [**mt ffig** 3 **fvel** 10]
?**fvel** 0 (las tortugas activas se detienen)

---

pi — comando

Significa **p**luma **i**nversa. Hace que la o las tortugas activas dibujen líneas a menos que se encuentren sobre una línea ya dibujada, en cuyo caso, la borra. Para volver al estado normal se debe ordenar **conpluma**.
Ejemplo:



?**ad** 60
?**pi at** 120

---

pintar — comando

Colorea un área cerrada de la pantalla con el color de la pluma de la o las tortugas activas. La tortuga debe estar adentro del área a pintar.

---

ponfig (número de figura) (código de figura) — comando

Significa **pon**er **fig**ura. Asigna a la figura indicada la forma definida por la lista (Ver Anexo C, El código de las figuras).
Ejemplo:

?**ponfig** 15 [1 1 1 1 1 1 1 255 255 1 1 1 1 1 1 1 128 128 128 128 128 128 128 255 255 128 128 128 128 128 128 128]
?**ffig** 15 **mt**

---

ponpri (objeto) (lista) — operación

Significa **pon**er **pri**mero. Responde con la lista que resulta de poner el primer argumento al comienzo de la lista dato.
Ejemplo:

?**mostrar ponpri** "aquí [me pongo a cantar]
[aquí me pongo a cantar]

---

ponúlt (objeto) (lista) — operación

Significa **pon**er **últ**imo. Responde con la lista que resulta de poner el primer argumento al final de la lista dato.
Ejemplo:

?**mostrar ponúlt** "solitaria [como el ave]
[como el ave solitaria]

---

pos — operación

Significa **pos**ición. Responde con una lista formada por las coordenadas x e y de la tortuga.
Ejemplo:

?**de** 30 **ad** 50
?**esc pos**
25 43,30127

---

pprop (nombre) (propiedad) (objeto) — comando

Significa **p**oner **prop**iedad. Asigna una propiedad y su valor a un nombre. El primer argumento es el nombre del objeto al cual se va a asociar una propiedad, el

segundo es el nombre de la propiedad y el tercero su valor.
Ejemplo:

?**pprop** "manzana "sabor "dulce

---

primero (objeto) operación
Responde con el primer elemento de su argumento.
Ejemplo:

?**esc primero** "Paula
P
?**esc primero** [bosque lago sierra]
bosque

---

primitiva? (nombre) operación
Responde **cierto** si su argumento es una primitiva; en caso contrario responde **falso**.
Ejemplo:

?**esc primitiva?** "primer
falso

---

primitivas comando
Lista en la pantalla todas las primitivas Logo.

---

producto (número) (número) operación
(**producto** (número) (número) ...)
Responde con el resultado de multiplicar sus argumentos.
Ejemplo:

?**esc** (producto 1 2 3 4)
24

---

proporción operación
Responde con la relación actual entre el paso vertical de la tortuga y el horizontal. El valor habitual es 1,375.
Ejemplo:

?**esc proporción**
1,375

---

punto [x y] comando
Pone un punto en el lugar indicado por su argumento.
Ejemplo:





?**punto** [– 10 50]

---

quién operación
Responde con la lista de las tortugas activas.
Ejemplo:
?**activar** [0 1 2 3]
?**esc quién**
0 1 2 3

---

rc (número) operación
Significa raíz cuadrada. Responde con la raíz cuadrada de su argumento.
Ejemplo:

?**esc rc** 2
1.4142135

---

reazar comando
Hace que **azar** responda la misma secuencia de números.
Ejemplo:

?**reazar repetir** 3 [**esc azar** 10]
1
1
6
?**reazar repetir** 3 [**esc azar** 10]
1
1
6

---

reciclar comando
Hace que Logo elimine de la memoria los objetos que ya no serán utilizados. Logo realiza automáticamente esta tarea cada vez que se ocupa toda la memoria, deteniendo momentáneamente la ejecución de los procedimientos. Se usa **reciclar** dentro de los procedimientos para que esta tarea se ejecute en los momentos más convenientes.

---

recordar (archivo) comando
Transfiere el contenido de un archivo de disco al espacio de trabajo. Aparece un mensaje en la pantalla a medida que Logo va definiendo los procedimientos.
Ejemplo:

?**recordar** "polígono

---

recordarc (archivo) comando
Significa **recordar** de **c**asete. Transfiere el contenido de un archivo de casete al espacio de trabajo. Es imprescindible que esté apretada la tecla de reproducción del grabador. Aparece un mensaje a medida que Logo va definiendo los procedimientos.



Ejemplo:

?**recordarc** "flores

---

recordardib (archivo) comando
Significa **recordar dib**ujo. Transfiere el contenido de un archivo de dibujos del disco a la pantalla.
Ejemplo:

?**recordardib** "ciudades

---

redondeo (número) operación
Responde con el número que resulta de redondear el argumento al entero más próximo.
Ejemplo:

?**esc redondeo** 1,7
2

---

remprop (nombre) (propiedad) comando

Significa **rem**over **prop**iedad. Elimina la propiedad indicada por el segundo argumento del nombre al cual está asociada.
Ejemplo:

?**remprop** "manzana "color.

---

repetir (número) (lista de instrucciones) comando
Cumple la lista de instrucciones tantas veces como indica su primer argumento.
Ejemplo:

?**repetir** 3 [**ad** 40 **de** 120]

---

respuesta, resp (objeto) comando
Detiene el procesamiento en que se encuentra y responde el objeto especificado por su argumento. Sólo se utiliza dentro de procedimientos.
Ejemplo:

para doble :n
respuesta 2 * :n
fin

---

resto (número) (número) operación
Responde con el resto de la división entera del primer argumento por el segundo.
Ejemplo:

?**esc resto** 10 7
3

---

rg comando

Significa **r**estaurar **g**ráficos. Restaura la pantalla de gráficos al estado que tiene cuando se inicia Logo. Es decir, ubica todas las tortugas en el centro, con rumbo y velocidad 0, les asigna la figura de tortuga, borra los demonios, pone el color de fondo 5 y el color de texto y de la pluma 15; activa la tortuga 0 y la hace visible.

---

ruido (tono) (amplitud) (duración) comando
Genera un ruido de las características indicadas por sus argumentos. Tono puede valer de 0 a 31, variando el ruido de más agudo a más grave. Amplitud puede valer 0 a 15, variando el volumen de más bajo a más alto. Duración puede valer de 0 a 255 y prolonga el ruido por duración/50 segundos.
Ejemplo:

?**ruido** 28 10 100

---

rumbo operación
Responde con la orientación medida en grados de la tortuga. Un **rumbo** 0 indica que la tortuga apunta hacia arriba; **rumbo** 90, que apunta hacia la derecha.
Ejemplo:

?**iz** 90 **esc rumbo**
270

---

sen (número) operación
Responde con el seno del ángulo dado como argumento.
Ejemplo:

?**esc sen** 30
0,5

---

si (predicado) (lista de instrucciones) comando
si (predicado) (lista de instrucciones) (lista de instrucciones)
Ejecuta la primera lista de instrucciones si el predicado es **cierto**; si es **falso** ejecuta la segunda lista de instrucciones o pasa al renglón siguiente si la lista no existe.
Ejemplo:

?**si vel** > 50 [**fvel** 0]

---

sinimpresora comando
Desactiva la impresora

---

sinpluma, sp comando
Hace que la o las tortugas activas no dibujen al desplazarse.
Ejemplo:

?**sp at** 30

---

sombrear comando
Repite en un área cerrada de la pantalla la figura de la tortuga activa. La tortuga debe estar adentro del área a sombrear.





sonido (canal) (frecuencia) (amplitud) (duración) comando
Genera un sonido de las características indicadas por sus argumentos. Se pueden generar hasta tres sonidos simultáneamente utilizando los tres canales disponibles (0, 1 y 2). Frecuencia puede valer de 28 a 4095, estableciendo la frecuencia del sonido. Amplitud puede valer de 0 a 15, variando el volumen de más bajo a más alto. Duración puede valer de 0 a 255 y prolonga el sonido por duración/50 segundos. (Ver en el Anexo A la Tabla de frecuencias y el Capítulo 7, Logo y la Música).
Ejemplo:

```
?sonido 0 880 10 120
```

suma (número) (número) operación
(**suma** (número) (número) ...)
Responde con la suma de sus argumentos.
Ejemplo:

```
?esc (suma 1 2 3 4)
10
```

tecla ? operación
Responde **cierto** si se ha pulsado una tecla y aún no se la ha leído; en caso contrario responde **falso.**
Ejemplo:

```
?para saludo
si tecla? = cierto [esc "adiós parar]
esc [pulsa cualquier tecla para parar]
saludo
fin
```

texto (nombre) operación
Responde con la lista de las instrucciones del procedimiento indicado por su argumento. Si el procedimiento tuviera variables, los nombres de éstas se encontrarán en el primer elemento de la lista de instrucciones; si no hubiera variables, el primer elemento será una lista vacía.
Ejemplo:

```
?mostrar texto "mensaje
[ [nombre] [esc :nombre] [esc "ganaste] ]
```

todas operación
Responde con la lista de los números de todas las tortugas, es decir, una lista de los números enteros de 0 a 29.
Ejemplo:

```
?activar todas mt
?cada [fvel quién / 2]
```

último (objeto) operación



Responde con el último elemento de su argumento.
Ejemplo:

```
?esc último "Paula
a
?esc último [bosque lago sierra]
sierra
```

vacía? (objeto) operación
Responde **cierto** si su argumento es una palabra o lista vacía. En caso contrario responde **falso**.
Ejemplo:

```
?esc vacía? [ ]
cierto
?esc vacía?"
cierto
```

velocidad, vel operación
Responde con la velocidad de la tortuga activa.
Ejemplo:

```
?esc vel
0
```

velx operación
Significa **vel**ocidad **x**. Responde con la componente x (horizontal) de la velocidad de la o las tortugas activas.
Ejemplo:

```
?frumbo 45 fvel 10
?esc velx
7,0710678
```

vely operación
Significa **vel**ocidad **y**. Responde con la componente y (vertical) de la velocidad de la o las tortugas activas.
Ejemplo

```
?frumbo 45 fvel 10
?esc vely
7,0710678
```

ventana comando
Permite que la tortuga se mueva fuera de los límites de la pantalla, que, en este caso, sólo muestra una parte del inmenso campo en que puede moverse la tortuga.
Ejemplo:

```
?ventana
```

?**ad** 200

versión comando
Imprime en la pantalla el código de la versión de Logo en uso.

visible? operación
Responde **cierto** si se ve la figura de la o las tortugas activas; en caso contrario responde **falso**.
Ejemplo:

?**ot esc visible?**
falso

vuelta comando
Hace que la tortuga, cuando sale por un borde, reaparezca por el lado opuesto con la misma dirección y sentido.
Ejemplo:

?**vuelta ad** 200

y (predicado) (predicado) operación
(**y** (predicado) (predicado) ...)
Responde **cierto** si todos sus argumentos son ciertos; si alguno es falso, responde **falso.**
Ejemplo:

?**si y vel** = 0 **rumbo** = 0 [**esc** "llegaste!]

Operaciones

(número) * (número) operación
Responde con el resultado de multiplicar ambos argumentos. Es similar a **producto.**
Ejemplo:

?**esc** 1 * 2 * 3 * 4
24

(número) + (número) operación
Responde con el resultado de sumar ambos argumentos. Es similar a **suma.**
Ejemplo:

?**esc** 1 + 2 + 3 + 4
10

(número) — (número) operación
Responde con el resultado de restar el segundo argumento al primero. Es similar a **dif.**
Ejemplo:



?**esc** 23456,7 – 12345,6
11111,1

(número) / (número) operación
Responde con el resultado de dividir el primer argumento por el segundo. Es similar a **cociente.**
Ejemplo:

?**esc** 875 / 5
175

(número) < (número) operación
Responde **cierto** si el primer argumento es menor que el segundo; en caso contrario, responde **falso.**
Ejemplo:

?**esc** 4 < 5
cierto

(objeto) = (objeto) operación
Responde **cierto** si los argumentos son iguales; en caso contrario, responde **falso**. Es similar a **igual?**
Ejemplo:

?**esc colorf** = 5
cierto

(número) > (número) operación
Responde **cierto** si el primer argumento es mayor que el segundo; en caso contrario, responde **falso.**
Ejemplo:

?**esc** 4 > 5
falso

Caracteres especiales

Los paréntesis, (o), se utilizan para modificar el orden que emplea Logo habitualmente para efectuar operaciones; en general, las funciones indicadas dentro de los paréntesis se ejecutan primero. Cuando realiza operaciones matemáticas, Logo sigue la convención usual de ejecutar primero las multiplicaciones y divisiones y luego las sumas y restas; este orden puede variarse usando paréntesis. Los paréntesis se usan además para aumentar la cantidad de argumentos que admiten algunas primitivas (Ver la primera parte de este capítulo).
Ejemplo:

?**esc** 2 * 6 + 7
9
?**esc** 2 * (6 + 7)
26



"(comillas)
Las comillas, ", cuando preceden inmediatamente a una palabra, indican a Logo que la palabra se utiliza por sí misma, es decir, que no se trata del nombre de un procedimiento o una variable.

Nota: Los números son palabras especiales, formadas solamente por dígitos, y no requieren comillas.
Ejemplo:

```
?esc "hola
hola
?esc 7
7
```

: (dos puntos)
Los dos puntos, :, precediendo inmediatamente a una palabra le indican a Logo que esa palabra es el nombre de una variable y que debe utilizar el contenido o valor de esa variable.
Ejemplo:

```
?hacer "ángulo 30
?esc :ángulo
30
?izquierda: ángulo
```

[ ] (corchetes)
Los corchetes, [ ], se utilizan en Logo para delimitar listas.
Ejemplo:

```
?hacer "saludo [buenas tardes]
?escribir :saludo
buenas tardes
```

Delimitadores
Se llaman delimitadores en Logo a algunos caracteres especiales, que se usan para separar palabras. El más común es el espacio, pero Logo considera también a los siguientes símbolos como delimitadores:



\ (barra invertida)
La barra invertida, \ , seguida de un carácter delimitador, le indica a Logo que debe considerar a éste como un carácter normal.
Ejemplo:

```
?esc [985-4482]
985 - 4482
?esc [985\-4482]
985-4482
```



Primitivas especiales

Las primitivas especiales ofrecen a los expertos la posibilidad de entrar en la "intimidad" de la máquina, ya que les permiten acceder a zonas de la memoria de la computadora que en general se encuentran protegidas. Hay que tener cuidado en su uso pues pueden llegar a alterar la lógica interna que regula el funcionamiento de la computadora. En general, estas órdenes sólo serán útiles a aquellas personas que tengan conocimientos de programación en lenguaje de máquina y también del diseño específico de la computadora.
Para diferenciarlas claramente de las restantes primitivas, los nombres de las primitivas especiales comienzan siempre con un punto.

.dentro (número de toma) — operación
Responde el contenido de la toma de entrada indicada por su argumento. Para determinar cuáles son las tomas de entrada disponibles se debe consultar la documentación técnica de la máquina.

.depositar (dirección) (byte) — comando
Escribe el segundo argumento en la posición de memoria indicada por el primero.

.examinar (dirección) — operación
Responde el valor decimal del byte ubicado en la posición de memoria especificada por su argumento.

.fuera (byte) (número de toma) — comando
Envía el valor de su primer argumento a la toma especificada por el segundo argumento.

.llamar (dirección) — comando
Inicia la ejecución de un procedimiento escrito en lenguaje de máquina cuyas instrucciones se encuentran a partir de la dirección especificada.



CAPÍTULO 9 MENSAJES DE PANTALLA

1. (procedimiento) **definido**
   Aparece al salir del Editor o cuando se recuerda un archivo de un disco o de cinta, indicando que no ha habido inconvenientes en definir el procedimiento.

2. **Número demasiado grande**
   Aparece cuando una operación aritmética da como resultado un número demasiado grande (mayor que 10,0e + 38) para ser procesado por Logo.

3. (simbolo) **es un primitiva**
   Este mensaje aparece cuando el usuario intenta:
   —Dar a un procedimiento el nombre de una primitiva.
   —Definir un nombre predefinido.

4. **Imposible editar del editor**
   Aparece al salir del Editor, si se ha usado dentro de él la orden **editar**.

5. **Imposible dividir por cero**
   Aparece cuando el valor del divisor es cero.

6. **No hay lugar en el disco**
   Indica que no hay lugar en el disco para guardar un nuevo archivo. Se debe elegir otro disco o reducir el tamaño de lo que se desea guardar, borrando los procedimientos innecesarios del espacio de trabajo.

7. **No se puede usar el disco**
   Indica que el disco no ha sido preparado para usar con la computadora MSX. Ver el Manual del Usuario de la unidad de disco.

8. **Ese archivo no existe**
   Aparece cuando se intenta **recordar** un archivo que no existe en el disco o cinta utilizados.

9. **Ese archivo ya existe**
Aparece cuando se intenta usar la orden **guardar** con el nombre de un archivo existente. Este mensaje aparece porque Logo no permite grabar sobre un archivo anterior sin que previamente se lo elimine con **boarchivo**.

10. **No hay espacio**
Aparece cuando se ha ocupado toda la memoria disponible de la máquina.

11. (símbolo) **ni cierto ni falso**
Aparece cuando el predicado de la orden **si** no es ni **cierto** ni **falso**.

12. **Faltan argumentos para** (procedimiento)
Aparece cuando Logo espera más argumentos que los que le han sido dados. Ver la definición de la función que se ha usado para obtener más información.

13. **Falta espacio para editar**
Aparece cuando se quieren ingresar al Editor demasiados procedimientos o procedimientos muy largos.

14. **La tortuga no está en la ventana**
Aparece cuando se ha dado la orden de **pintar** o **sombrear**, con la tortuga fuera de la pantalla.

15. **No se cómo hacer** (símbolo)
Aparece cuando el procedimiento solicitado por el usuario no existe en la memoria de Logo. Puede suceder por errores de ortografía o porque falta definir el procedimiento.

16. **No existe ninguna cosa llamada** (símbolo)
Aparece cuando el símbolo solicitado por el usuario no existe en la memoria de Logo. Puede suceder por errores de ortografía o porque falta definir el nombre.

17. **Paréntesis mal colocado**
Aparece cuando faltan cerrar paréntesis o se han cerrado sin abrirlos.

18. **No sé qué hacer con** (símbolo)
Indica que falta una orden. Agregar la instrucción faltante.

19. (procedimiento) **no acepta** (símbolo) **como argumento**
Aparece cuando se trata de dar a un procedimiento un argumento que no le corresponde (por ejemplo, **adelante** "María).

20. (procedimiento) **no responde a** (procedimiento)
Aparece cuando se utiliza como argumento un comando en lugar de una operación.

21. **Pocos ítems en** (símbolo)
Aparece cuando el número dado como primer argumento a **ítem** es mayor que la cantidad de objetos del segundo argumento.

22. **Hay problemas con el dispositivo**
Indica que existe alguna falla en el dispositivo de entrada o de salida (por ejemplo, que la impresora está apagada o no hay disco en la unidad de disco).

23. **No se puede usar en nivel superior**
Aparece cuando se trata de usar en el nivel superior una orden que está permitida sólo dentro de procedimientos.

24. **Demasiado complejo para pintar o sombrear**
Indica que en la pantalla hay demasiadas figuras o dibujos para poder pintar o sombrear.

25. **Demasiados demonios**
Indica que se ha superado el número máximo de demonios que es posible establecer simultáneamente.

26. **Demasiados argumentos**
Aparece cuando se intenta definir un procedimiento con un número demasiado grande de argumentos.

27. **Parada!!!**
Aparece cuando se detiene la ejecución de un procedimiento con **CTRL-STOP**

28. **Bloques disponibles**
Aparece cuando se solicita el directorio de un disco. Indica el espacio libre (en bloques) que hay en el disco.

29. **Bloques ocupados**
Aparece cuando se solicita el directorio de un disco. Indica el espacio ocupado (en bloques) que hay en el disco.

ANEXO A

TABLAS DE COLORES, FIGURAS, ACONTECIMIENTOS, CARACTERES Y FRECUENCIAS MUSICALES

COLORES

MSX-Logo ofrece 16 tonos utilizables para el color de las tortugas, el color con que dibujan, el color del fondo y el color del texto. (Ver la Tabla de colores más abajo).

Al comenzar, tanto las tortugas como sus plumas llevan el color blanco (color número 15) y el fondo de la pantalla es azul claro (color número 5).

El cambio de los colores del fondo, de las tortugas y de sus plumas puede dar lugar a efectos visuales asombrosos. La orden **fcolorf** (**f**ijar **color**) se emplea para cambiar el color de las tortugas y, la orden **fcolorp** (**f**ijar **color p**luma) para cambiar el de la pluma. La orden **fcolorf** (**f**ijar **color f**ondo) varía el color del fondo y **cambiar.color** modifica el color de todo gráfico o texto que aparezca en la pantalla.

Nota: La tortuga o su trazo resultarán invisibles si ella o su pluma son transparentes (color 0) o del mismo tono que el fondo.

TABLA DE COLORES

Número	Color	Número	Color
0	transparente	8	rosa
1	negro	9	rosa claro
2	verde medio	10	amarillo
3	verde claro	11	amarillo claro
4	azul oscuro	12	verde oscuro
5	azul claro	13	magenta
6	rojo	14	gris
7	celeste	15	blanco

FIGURAS DE LAS TORTUGAS

Al comenzar Logo, todas las tortugas tienen la misma forma, que es la de una tortuga vista desde arriba. Sin embargo, pueden adoptar cualquiera de las 60 figuras numeradas de 0 a 59 disponibles en MSX-Logo. La orden **ffig** asigna una figura a una tortuga. Se pueden crear nuevas figuras o modificar las existentes usando el Editor de figuras. (Ver el Anexo B). MSX-Logo provee las siguientes figuras:

TABLA DE FIGURAS

Número	Figura
0	círculo
1	corazón
2	gato
3	perro
4	camión
5	cohete
6	pared
7	helicóptero
8	locomotora
9	vagón
10-35	cuadro
36-59	figura de tortuga con distintas orientaciones

DEMONIOS

El demonio es un tipo especial de medio de programación; su funcionamiento es independiente de todo programa en ejecución.
Las primitivas que crean demonios son **al.contacto** y **cuando**. Los demonios creados por **al.contacto** observan si se produce una colisión entre tortugas. Los demonios de **cuando** actúan si ocurre alguno de los acontecimientos de la tabla que sigue.

Tabla de Acontecimientos

Número	Acontecimiento
0	transcurre alrededor de un segundo
1	pulsación de una tecla
2	pulsación del botón del mando 1
3	pulsación del botón del mando 2
4	movimiento de la palanca del mando 1
5	movimiento de la palanca del mando 2

Cuando se produce el acontecimiento, el demonio interrumpe los procedimientos en ejecución y procesa una lista propia de instrucciones. Una vez que éstas se han ejecutado, el programa anterior continúa desde el punto de interrupción. Los demonios permanecen activos, en espera de un suceso o colisión entre tortugas, hasta que son eliminados con la orden **bods** (**bo**rrar **d**emonio**s**). Otra forma de eliminarlos es especificando una lista de instrucciones vacía.

TABLA DE CARACTERES

Las ordenes **ascii** y **carácter** responden valores de acuerdo con el código ASCII. ASCII es la abreviatura de American Standard Code of Information Interchange (Código Americano Standard para el Intercambio de Información). En este código, muy difundido en el área de las comunicaciones por computadoras, cada símbolo está representado por un número.
El siguiente cuadro muestra las equivalencias entre cada símbolo disponible en Logo y su correspondiente número ASCII.

0	→	31	┼	62	>	93	]	124	|	155	¢
1	☺	32	ESP	63	?	94	^	125	}	156	£
2	☻	33	¡	64	@	95	_	126	~	157	¥
3	♥	34	"	65	A	96	`	127	▲	158	Pt
4	♦	35	#	66	B	97	a	128	Ç	159	ƒ
5	♣	36	$	67	C	98	b	129	ü	160	á
6	♠	37	%	68	D	99	c	130	é	161	í
7	•	38	&	69	E	100	d	131	â	162	ó
8	◘	39	'	70	F	101	e	132	ä	163	ú
9	TAB	40	(	71	G	102	f	133	à	164	ñ
10	◙	41	)	72	H	103	g	134	å	165	Ñ
11	♂	42	*	73	I	104	h	135	ç	166	ª
12	♀	43	+	74	J	105	i	136	ê	167	º
13	CR	44	,	75	K	106	j	137	ë	168	¿
14	♫	45	-	76	L	107	k	138	è	169	⌐
15	☼	46	.	77	M	108	l	139	ï	170	¬
16	┼	47	/	78	N	109	m	140	î	171	½
17	┴	48	0	79	O	110	n	141	ì	172	¼
18	┬	49	1	80	P	111	o	142	Ä	173	¡
19	┤	50	2	81	Q	112	p	143	Å	174	«
20	├	51	3	82	R	113	q	144	É	175	»
21	┼	52	4	83	S	114	r	145	æ	176	Ã
22	│	53	5	84	T	115	s	146	Æ	177	ã
23	─	54	6	85	U	116	t	147	ô	178	Ĩ
24	┌	55	7	86	V	117	u	148	ö	179	ĩ
25	┐	56	8	87	W	118	v	149	ò	180	Õ
26	└	57	9	88	X	119	w	150	û	181	õ
27	┘	58	:	89	Y	120	x	151	ù	182	Ũ
28	╳	59	;	90	Z	121	y	152	ÿ	183	ũ
29	╱	60	<	91	[	122	z	153	Ö	184	Ĳ
30	╲	61	=	92	\	123	{	154	Ü	185	ĳ

186	¾	198	▏	210	⧗	222	▐	234	Ω	246	÷
187	~	199	▞	211	▘	223	▀	235	δ	247	≈
188	◊	200	▊	212	▗	224	α	236	⁃	248	∘
189	‰	201	▕	213	▝	225	β	237	∅	249	•
190	¶	202	▊	214	▖	226	Γ	238	∈	250	•
191	§	203	▨	215	▩	227	η	239	∩	251	√
192	▁	204	▧	216	⟁	228	Σ	240	≡	252	ⁿ
193	▚	205	▼	217	ǂ	229	σ	241	±	253	²
194	▆	206	▲	218	ω	230	μ	242	≥	254	⎕
195	▔	207	▶	219	█	231	ϒ	243	≤	255	
196	⊟	208	◀	220	▄	232	ɵ	244	⌠		
197	▆	209	⧓	221	▌	233	Θ	245	⌡		

TABLA DE FRECUENCIAS

La siguiente tabla muestra las frecuencias (redondeadas a enteros) de siete octavas de la escala temperada (un semitono entre notas).

FRECUENCIA	NOTA
28	LA
29	LA #,SIb
31	SI
33	DO
35	DO #,REb
37	RE
39	RE #,MIb
42	MI
44	FA
47	FA #,SOLb
49	SOL
52	SOL #,LAb
55	LA
59	LA #,SIb
62	SI
66	DO
70	DO #,REb
74	RE
78	RE #,MIb
83	MI
88	FA
93	FA #,SOLb
98	SOL
104	SOL #,LAb
110	LA
117	LA #,SIb
123	SI
131	DO

FRECUENCIA	NOTA
139	DO #,REb
147	RE
156	RE #,MIb
165	MI
175	FA
185	FA #,SOLb
196	SOL
208	SOL #,LAb
220	LA
233	LA #,SIb
247	SI
262	DO (DO central)
277	DO #,REb
294	RE
311	RE #,MIb
330	MI
349	FA
370	FA # .SOLb
392	SOL
415	SOL #,LAb
440	LA
466	LA #,SIb
494	SI
523	DO
554	DO #,REb
587	RE
622	RE #,MIb
659	MI

698	FA	1661	SOL #,LAb
740	FA #,SOLb	1760	LA
784	SOL	1864	LA #,SIb
831	SOL #,LAb	1976	SI
880	LA	2094	DO
932	LA #,SIb	2218	DO #,REb
988	SI	2350	RE
1047	DO	2489	RE #,MIb
1109	DO #,REb	2638	MI
1175	RE	2794	FA
1245	RE #,MIb	2960	FA #,SOLb
1319	MI	3135	SOL
1397	FA	3322	SOL #,LAb
1480	FA #,SOLb	3520	LA
1568	SOL	3728	LA #,SIb

ANEXO B

TECLAS PARA EDICIÓN DE TEXTO Y FIGURAS

Editor Logo

El Editor Logo se utiliza para definir y editar procedimientos y variables, introduciéndolos después en el espacio de trabajo. Se ingresa al Editor con las órdenes **para**, **editar** y **edns** (**ed**itar **n**ombre**s**).
Cuando se trabaja en el Editor, se pueden utilizar las siguientes teclas de edición:

Funciones de edición:

Nota: El asterisco (*) señala las teclas o combinaciones de teclas cuyo funcionamiento es válido tanto dentro como fuera del Editor Logo.

* →	Mueve el cursor un espacio a la derecha.
* ←	Mueve el cursor un espacio a la izquierda.
* ↓	Baja el cursor a la línea siguiente.
* ↑	Sube el cursor a la línea anterior.
* **CTRL - A**	Desplaza el cursor al principio de la línea.
* **CTRL - E**	Lleva el cursor al final de la línea.
* **CTRL - T**	Lleva el cursor al principio del primer renglón del editor.
* **CTRL - B**	Lleva el cursor al final del último renglón del editor.

Inserción y supresión de caracteres y líneas

* **RETURN**	Crea una nueva línea; el cursor, junto con el texto que se encuentra a su derecha, se mueve al principio de la nueva línea de pantalla. Fuera del Editor, al pulsar la tecla **RETURN** se indica a Logo que ejecute lo que se introdujo por teclado.

	INS	Abre una nueva línea en la posición del cursor, pero sin moverlo.
*	BS	Mueve el cursor un lugar hacia la izquierda, borrando el carácter existente.
*	DEL	Borra el carácter bajo el cursor.
*	CTRL · K	Borra el texto desde la posición del cursor hasta el final de la línea. Este texto pasa a la memoria intermedia de supresión, cuya capacidad permite alojar 256 caracteres.
*	CTRL · Y	Inserta en la posición del cursor una copia del texto contenido en la memoria intermedia de supresión.

Salida del Editor

	ESC	Significa ESCapar. Sale del Editor y agrega al espacio de trabajo todas las definiciones del procedimiento y variables ingresadas o modificadas. Se ejecutan todas las instrucciones no incluidas en procedimientos.
*	CTRL · STOP	Abandona el Editor, ignorando todas las modificaciones que se hayan introducido durante la edición. Si se ha corregido un procedimiento, la definición del mismo quedará igual que antes de comenzar la edición Fuera del Editor, el uso de la combinación de teclas **CTRL · STOP** detiene la ejecución de todos los procedimientos.

Editor de figuras

La orden **edfig** sirve para colocar una figura en el Editor de figuras, a fin de modificarla o crear una nueva. El Editor de figuras es una cuadrícula de 16 por 16 casillas. Las casillas llenas (negras) definen la figura. El cursor (rectángulo blanco parpadeante) indica el punto donde se está trabajando. Para crear figuras en el Editor de figuras se utilizan las siguientes teclas o combinaciones de ellas.

→	Lleva el cursor una casilla a la derecha.
←	Lleva el cursor una casilla a la izquierda.
↑	Sube el cursor una casilla.
↓	Baja el cursor una casilla.



Barra de espaciado	Llena una casilla vacía o deja en blanco una casilla llena.
CTRL · K (o CLS/HOME)	Borra toda la figura, dejando la cuádrícula en blanco.
CTRL · Y	Restaura la figura existente en el Editor de figuras antes de comenzar las modificaciones.
ESC	Sale del Editor de figuras, guardando los cambios introducidos.
CTRL · STOP	Abandona el Editor de figuras sin guardar ninguno de los cambios realizados.

Teclas especiales

Barra de espaciado	Imprime un carácter muy importante, aunque invisible, llamado espacio. Logo utiliza los espacios para separar las palabras. Por ejemplo, interpreta holaamigos como una sola palabra y hola amigos como dos.
Tecla **SHIFT**	Si se pulsan simultáneamente la tecla **SHIFT** y la tecla de un carácter alfabético, se escribirá el carácter en mayúscula. También se usa la tecla **SHIFT** para escribir el símbolo indicado en la parte superior de algunas teclas. Por ejemplo pulsando conjuntamente **SHIFT** y la tecla **1**, se imprime en la pantalla un ! (signo de admiración).
Tecla **CTRL**	La pulsación conjunta de **CTRL** con otra tecla produce diversas funciones especiales. Por ejemplo, si se pulsa la tecla **A** manteniendo pulsada la tecla **CTRL**, el cursor se desplazará al comienzo de la línea en la que se encuentra. En la primera parte de este Anexo se indican las funciones que se encuentran disponibles.
Tecla **TAB**	La tecla de tabulación se utiliza para dar un formato a los procedimientos (por formato se entiende el aspecto de un procedimiento). La tecla **TAB** facilita escribir procedimientos de modo que su lectura sea más fácil y comprensible, separando líneas largas en varios renglones. Cuando se la pulsa, Logo escribe espacios hasta el final del renglón y ubica al cursor en el siguiente; una flecha de continuación aparece al final del renglón de la pantalla. Por ejemplo:

```
para probar :x :y
si :x > :y                 →   (pulsar la tecla TAB)
   [escribe [mayor]]       →   (agregar espacios, escribir y luego pulsar la
                                tecla TAB)
   [escribe [menor]]           (agregar espacios y luego escribir)
fin
```


ANEXO C

EL CÓDIGO DE LAS FIGURAS

La computadora emplea números para recordar las configuraciones que definen las figuras. Se llama código de una figura a una lista de 32 números enteros que, de acuerdo con una cierta clave que Logo conoce, representan la forma de la figura.

Si bien mediante el Editor de figuras es posible crear cualquier forma sin necesidad de conocer o utilizar su código, a continuación se hace una breve reseña de su significado y de cómo usarlo para guardar y recordar figuras en disco o cinta.

La orden privativa **obfig** (**ob**tener código de **fig**ura) nos permite conocer el código de las figuras ya definidas, y, si así se desea, guardarlas en variables. Por ejemplo:

?mostrar obfig 10
[255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255]
?hacer "vieja.figura **obfig** 10

La orden **ponfig** tiene una función recíproca a la de **obfig**: da a la figura indicada, la forma que corresponde a su lista dato. Por ejemplo:

?ponfig 20 [170 85 170 85 170 85 170 85 170 85 170 85 170 85 170 85 170 85 170 85
170 85 170 85 170 85 170 85 170 85 170 85]

creará una figura, la 20, formada por puntos alternativamente llenos y vacíos. Compruébelo con **edfig** 20.

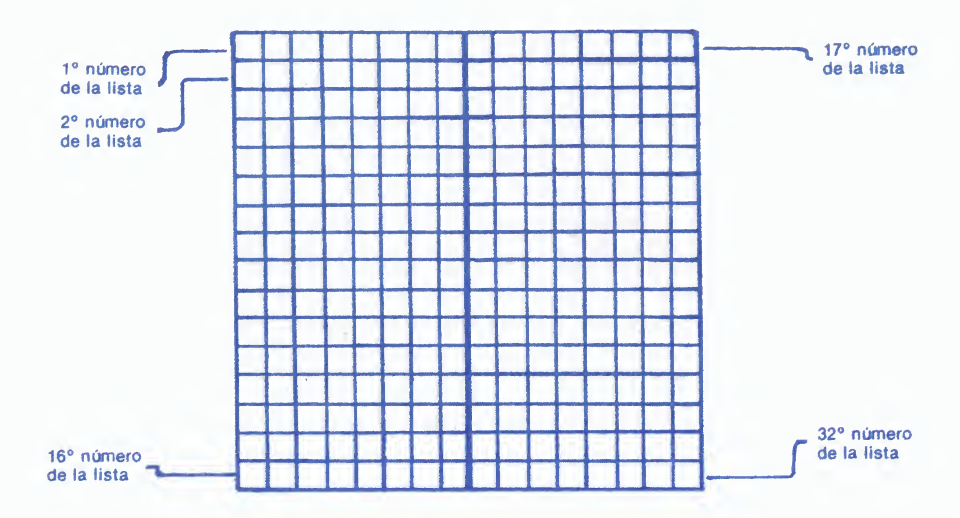
Para producir este código, Logo divide la grilla de 16 x 16 puntos que define las figuras, en dos mitades, la izquierda y la derecha. Cada mitad está formada por 16 renglones de 8 puntos; cada uno de los 32 números de la lista representan a cada uno de estos renglones.

La clave surge de asignar a cada punto lleno de un renglón un número 1, y un 0 a los puntos vacíos. Por ejemplo:

10101010 equivale a ■□■□■□■□
01010101 equivale a □■□■□■□■

10101010 es un número binario (es decir, que está expresado sólo con los dígitos 0 y 1) y su equivalente en el sistema decimal es el 170; el de 01010101 es 85.





Nota: El equivalente decimal del número binario abcdefgh es:
a ∗ 128 + b ∗ 64 + c ∗ 32 + d ∗ 16 + e ∗ 8 + f ∗ 4 + g ∗ 2 + h
Ejemplo:
El equivalente de 10101010 es
1 ∗ 128 + 0 ∗ 64 + 1 ∗ 32 + 0 ∗ 16 + 1 ∗ 8 + 0 ∗ 4 + 1 ∗ 2 + 0 =
128 + 32 + 8 + 2 = 170

Como guardar y recordar figuras

Para guardar figuras en el disco o en la cinta, es necesario almacenarlas en variables. Por ejemplo, si las figuras 14 y 15 tienen la forma de un árbol y de una flor, las siguientes órdenes:

?hacer "fig.árbol **obfig** 14
?hacer "fig.flor **obfig** 15

crearán las variables f.árbol y f.flor y les asignarán el valor del código de las figuras 14 y 15. Si se da ahora la orden **guardar**, estas variables, así como todo el espacio de trabajo, se grabarán en el disco (o en la cinta, si se ordena **guardarc**). Cuando se recuerde el archivo, Logo creará nuevamente las variables f.árbol y f.flor. Para restituir a las figuras 14 y 15 sus viejas formas, bastará ordenar:

?ponfig 14 :fig.árbol
?ponfig 15 :fig.flor



ANEXO D

LAS ÓRDENES DE LA TORTUGA ESPACIAL

Ordenes básicas (*)

Movimientos:

andar

virar
cabecear
rolar

Para iniciar:

tri

Para hacer aparecer o desaparecer la triada:

triada

Ordenes abreviadas (**)

aad
aat
vde
viz
rde
riz
cad
cat

Otras órdenes auxiliares (***)

distancia3d
hacia3d

(*) Ordenes incorporadas en el archivo TRI.
(**) Ordenes incorporadas en el archivo BREVES.
*(***) Ordenes incorporadas en el archivo TRIAUX. Ref.: **El uso artesanal de las computadoras,** Horacio C. Reggini, Buenos Aires, Galápago, Dist. Emecé (en publicación).*

pos3d
rumbo3d

fpos3d
frumbo3d

Ordenes básicas

andar (número) comando
Mueve la tortuga tridimensional en la dirección longitudinal la distancia indicada por su argumento, sin que varíe su plano ni su orientación, al igual que en el caso de la tortuga plana clásica.
Ejemplo:
?**andar** 50

cabecear (número) comando
Gira la tortuga tridimensional según su eje transversal el ángulo indicado por su argumento.
Ejemplo:
?**cabecear** 60

rolar (número) comando
Gira la tortuga tridimensional según su eje longitudinal el ángulo indicado por su argumento.
Ejemplo:
?**rolar** 60

tri comando
Prepara la escena para la realización de dibujos tridimensionales. Borra la pantalla y deja la tortuga lista para dibujar, en estado **visible** y en el modo **ventana**.

triada comando
Crea la triada y hace que aparezca —si no es visible— o desaparezca —si es visible— en la pantalla. La triada está formada por tres ejes que indican la orientación de la tortuga tridimensional en el espacio.

virar (número) comando
Gira la tortuga tridimensional en su propio plano, el ángulo indicado por su argumento, de una forma similar a las órdenes **derecha** e **izquierda** de la tortuga plana clásica.
Ejemplo:
?**virar** 90

Ordenes abreviadas

aad (número) comando
Significa andar **ad**elante. Equivale a **andar**.

aat (número) comando
Significa **a**ndar **at**rás. Equivale a **andar** con argumento negativo.

cad (número) comando
Significa **c**abecear **ad**elante. Equivale a **cabecear**.

cat (número) comando
Significa **c**abecear **at**rás. Equivale a **cabecear** con argumento negativo.

rde (número) comando
Significa **r**olar a la **de**recha. Equivale a **rolar**.

riz (número) comando
Significa **r**olar a la **iz**quierda. Equivale a **rolar** con argumento negativo.

vde (número) comando
Significa **v**irar a la **de**recha. Equivale a **virar** con argumento negativo.

viz (número) comando
Significa **v**irar a la **iz**quierda. Equivale a **virar**.

Otras órdenes auxiliares

distancia3d [x y z] operación
Responde con la distancia, medida en pasos de tortuga, entre la posición actual de la tortuga tridimensional y el punto indicado por el argumento.

Ejemplo:
?**tri esc distancia3d** [50 50 50]
86,60254

fpos3d [x y z] comando
Mueve la tortuga a la posición especificada por su argumento sin variar su orientación. Si la tortuga está **conpluma**, dejará un rastro al moverse.

Ejemplo:
?**fpos3d** [40 50 60]

frumbo3d [[c11 c12 c13] [c21 c22 c23] [c31 c32 c33]] comando
Impone a la tortuga tridimensional el rumbo indicado por su lista argumento; cada sublista de la lista argumento debe estar formada por los tres cosenos directores de cada uno de los ejes que definen la nueva orientación de la tortuga.

Ejemplo:
?**frumbo3d** [[0 1 0] [– 1 0 0] [0 0 1]]

hacia3d [x y z] operación

Responde con la lista de cosenos directores de los tres ejes que tendría la tortuga si estuviera mirando hacia el lugar especificado por el argumento. El eje longitudinal de la tortuga es determinado por la línea recta que va de la posición presente de la tortuga a la posición dada. Por convención, el eje transvérsal es horizontal y el eje perpendicular se dirige hacia arriba.

Ejemplo:
?tri esc hacia3d [45 45 45]
[0,57735027 0,57735026 0,57735027] [0,70710678 0 – 0,70710678] [0,47140452 0,81649658 – 0,40824829]

pos3d operación
Responde con una lista formada por las coordenadas x, y, z de la tortuga.

Ejemplo:
?tri andar 30 **virar** 40 **andar** 60
?esc pos3d
– 38,567256 75,962666 0

rumbo3d operación
Responde con el rumbo de la tortuga en el espacio como una lista de listas; cada sublista está formada por los tres cosenos directores de cada uno de los ejes de la tortuga.

Ejemplo:
?mostrar rumbo3d
[[0 1 0] [– 1 0 0] [0 0 1]]

Notas sobre el uso del sistema tridimensional

1) Archivos

Los procedimientos del sistema tridimensional se encuentran incorporados a los archivos TRI, BREVES y TRIAUX, cuyas características se explican a continuación:

Archivo TRI: contiene las órdenes básicas. Estas órdenes son las necesarias para mover la tortuga tridimensional según la geometría natural intrínseca de la tortuga. Este archivo incorpora también las órdenes que dibujan la triada.

Archivo BREVES: contiene las abreviaturas de las órdenes tridimensionales, las que incluyen en su nombre la indicación del sentido del movimiento.

Archivo TRIAUX: contiene las órdenes auxiliares tridimensionales similares a las órdenes disponibles para la tortuga bidimensional **distancia**, **hacia**, **fpos**, **frumbo**, **pos** y **rumbo**.

2) Procedimientos y variables

El sistema tridimensional comprende, además de las órdenes básicas y auxiliares vistas, los siguientes procedimientos internos:

&ejes.triada
&graficar
&reemplazar
&hacia3dA
&hacia3dB
&medio

y las siguientes variables internas:

:&c			variable auxiliar
:&c11	:&c12	:&c13	cosenos directores
:&c21	:&c22	:&c23	de los ejes
:&c31	:&c32	:&c33	intrínsecos
:&f			distancia focal
:&s			variable auxiliar
:&s11	:&s12	:&s13	
:&s21	:&s22	:&s23	variables auxiliares
:&s31	:&s32	:&s33	
:&t11	:&t12	:&t13	
:&t21	:&t22	:&t23	variables auxiliares
:&t31	:&t32	:&t3	
:&x			coordenadas
:&y			de la tortuga
:&z			en el espacio

Se ha antepuesto el símbolo poco común **&** a los nombres de variables y de procedimientos, de manera que no coincidan con los empleados comúnmente por los usuarios.

ANEXO E
COMPARACIÓN ENTRE MSX-LOGO Y OTRAS VERSIONES DE LOGO

A fin de facilitar el uso de procedimientos escritos en Logo según versiones anteriores, en este anexo se indican las diferencias entre MSX-Logo y las versiones utilizadas en la computadora TEXAS TI99/4A y en el libro Alas para la Mente.
Se debe tener presente que en MSX-Logo las primitivas se escriben siempre en minúscula.
En algunos casos, fue necesario modificar la traducción de las palabras primitivas para adecuarlas a los cambios introducidos en la versión inglesa LCSI.
El final de este anexo contiene una lista comparativa de las primitivas de ambas versiones. Las principales modificaciones son:

- La puntuación decimal se indica con coma (,) en lugar de punto (.).

- Llamamos ahora "tortugas" a los actores; por ello se ha cambiado la palabra **:TODOS** por su femenino **todas** (que es en MSX-Logo una orden de operación en lugar del contenido de una variable). Todas las tortugas en la versión MSX-Logo pueden dejar trazo al moverse.

- **HACERFIGURA** se ha reemplazado por **edfig** (**ed**itar **fig**ura), siguiendo el criterio de la versión inglesa de agrupar una serie de primitivas bajo el concepto de edición.

- Todas las primitivas que en inglés comienzan con **set**, empiezan en la versión castellana con la letra **f** (inicial de **fijar**). Así resultan **fcolorf** (fijar **color f**ondo), **fcolor** —para dar color a la figura de las tortugas—, **frumbo, fcolorp** (fijar **color p**luma), etc. La primitiva que indica la figura de la tortuga pasa a ser ahora **ffig** (fijar **fig**ura) en lugar de **LLEVAR.**

- La impresión de textos en la pantalla, que se hacía antes con la orden **IMPRIMIR**, se realiza ahora con la primitiva **escribir**. La anterior primitiva **ESCRIBIR** es reemplazada por **escribirs**; esta orden permite escribir seguido en un mismo renglón, es decir, deja el cursor a continuación del texto escrito sin pasar al renglón siguiente.

- Las órdenes que responden con las coordenadas de las tortugas se indican ahora con **coorx** y **coory** en lugar de **XCOOR** e **YCOOR**; con el mismo criterio, se usan **velx** y **vely** en lugar de **XVEL** e **YVEL**.
- Los operadores lógicos, que antes eran **AMBOS** y **UNOUOTRO**, se han traducido como **y** y **o**, siguiendo el criterio de la versión inglesa donde en lugar de **BOTH** y **EITHER** se emplean ahora **and** y **or**.
- La primitiva **ECARACTER** (ESCRIBIR **CARACTER**), que era una orden de comando que escribía la letra correspondiente en la pantalla, ha sido reemplazada por la orden de operación **carácter** que responde con la letra correspondiente al número dado como argumento.
- Se han preferido las palabras más expresivas **congelar, descongelar** y **activar**, en lugar de **DETENER, SEGUIR** y **DECIR**.
- La primitiva **ORACION** se ha modificado, adoptándose la palabra más breve **frase**.
- Las órdenes de listas, **ENTP** y **ENTU**, se traducen ahora como **ponpri** y **ponúlt**, incluyéndoselas dentro del grupo de primitivas que comienzan con el verbo poner.
- Las abreviaturas adoptadas para las primitivas **adelante, atrás, derecha** e **izquierda**, son respectivamente **ad, at, de** e **iz**, o sea, las dos primeras letras de cada palabra, en lugar de la primera y la última como era en las versiones anteriores. La abreviatura de **respuesta** se ha modificado, adoptándose la más común de **resp** en lugar de **RA**, que seguía el criterio antiguo.
- La anterior primitiva **LC?**, que respondía con los predicados **cierto** o **falso** según se hubiese pulsado o no una tecla, se ha reemplazado por **tecla?** adecuándola a la nueva nomenclatura inglesa **keyp**.

Nota: n. e. indica que no existe esa primitiva.
* indica a primitivas cuya traducción es distinta.
\# indica a primitivas cuya función no es exactamente igual en cada versión.

Listado comparativo de palabras primitivas:

		U.S.A.	VERSION NUEVA	VERSION TI99/4A
1		(	(	=
2		)	)	=
3		*	*	=
4		+	+	=
5		−	−	
6		.call	.llamar	n. e.
7		.deposit	.depositar	n. e.
8		.examine	.examinar	n. e.
9		.in	.dentro	n. e.
10		.out	.fuera	n. e.
11		/	/	=
12		<	<	=
13		=	=	=
14		>	>	=
15	*	all	todas	:TODOS
16		and	y	AMBOS
17		arctan	arctan	n. e.
18	*	ascii	ascii	NUMCARACTER
19		ask	pedir	n. e.
20		back	atrás	ATRAS
21		bf	mp	MP
22	*	bg	colorf	FONDO
23		bk	at	AS
24		bl	mu	MU
25		butfirst	menosprimero	MENOSPRIMERO
26		butlast	menosúltimo	MENOSULTIMO
27		buttonp	botón?	n. e.
28		cg	bg	n. e.
29		change.colour	cambiar.color	n. e.
30	#	char	carácter	ECARACTER
31		clean	limpiar	n. e.
32	#	cload	recordarc	RECORDAR
33		colour	color	COLOR
34		colour.over	color.debajo	n. e.
35		copydef	copiardef	n. e.
36		copysh	copiarfig	n. e.
37		cos	cos	n. e.
38	*	count	cuenta	LONGITUD
39		cs	bp	BP
40	#	csave	guardarc	GUARDAR
41		ct	bt	n. e.
42		cursor	cursor	n. e.
43		define	definir	DEFINIR
44		definedp	definido?	n. e.
45		diff	dif	DIFERENCIA
46		dir	dir	n. e.
47		distance	distancia	n. e.
48		dot	punto	PUNTO
49		each	cada	CADA
50		ed	ed	n. e.
51	*	edit	editar	REVISAR
52		edns	edns	n. e.
53		emptyp	vacía?	n. e.

		U.S.A.	VERSION NUEVA	VERSION TI 99/4A
54		end	fin	FIN
55		equalp	igual?	IGUAL?
56		er	bo	n. e.
57	*	erall	botodo	n. e.
58		erase	borrar	BORRAR
59		erd	bods	n. e.
60		erf	boarchivo	n. e.
61	#	ern	bon	BORRAR
62		erns	bons	n. e.
63		erprops	boprops	n. e.
64		erps	bops	n. e.
65	*	es	edfig	HACERFIGURA
66		false	falso	FALSO
67		fd	ad	AE
68		fill	pintar	n. e.
69		first	primero	PRIMERO
70		forward	adelante	ADELANTE
71		fput	ponpri	ENTP
72	*	freeze	congelar	DETENER
73		getsh	obfig	n. e.
74		gprop	obprop	n. e.
75		heading	rumbo	RUMBO
76		home	centro	CENTRO
77		ht	ot	OT
78		if	si	SI
79		int	ent	n. e.
80		item	item	n. e.
81		joy	mando	MANDO
82		joyp	mando?	n. e.
83		keyp	tecla?	LC?
84		last	último	ULTIMO
85		left	izquierda	IZQUIERDA
86		list	lista	n. e.
87		listp	lista?	n. e.
88		load	recordar	RECORDAR
89		loadpic	recordardib	n. e.
90		lput	ponúlt	ENTU
91		lt	iz	IA
92		make	hacer	HACER
93		minus	minus	n. e.
94		name	nombrar	LLAMAR
95		namep	nombre?	COSA?
96	#	noise	ruido	SONIDO

		U.S.A.	VERSION NUEVA	VERSION TI 99/4A
97		noprinter	sinimpresora	n. e.
98		not	no	NO
99		numberp	número?	NUMERO?
100		on.touch	al.contacto	n. e.
101		op	resp	RA
102		or	o	UNOUOTRO
103		output	respuesta	RESPUESTA
104		pc	colorp	n. e.
105		pd	cp	CP
106		pe	pb	PB
107		pendown	conpluma	CONPLUMA
108		penup	sinpluma	SINPLUMA
109		plist	listaprops	n. e.
110	#	po	imp	EI
111	*	poall	imtodo	ET
112		pods	imds	n. e.
113		pofile	imarchivo	n. e.
114	*	pons	imns	EN
115		pops	imps	n. e.
116	*	pos	pos	DONDE
117	*	pots	imts	EP
118		pprop	pprop	n. e.
119		pps	improps	n. e.
120		pr	esc	n. e.
121		primitivep	primitiva?	n. e.
122		primitives	primitivas	.SOCORRO
123		print	escribir	IMPRIMIR
124	*	printer	conimpresora	SALIDAIMPRESORA
125		product	producto	PRODUCTO
126		pu	sp	SP
127		putsh	ponfig	n. e.
128		px	pi	PI
129		quotient	cociente	COCIENTE
130		random	azar	AZAR
131		rc	lc	LC
132	#	recycle	reciclar	.BM
133		remainder	resto	n. e.
134		remprop	remprop	n. e.
135		repeat	repetir	REPETIR
136		rerandom	reazar	n. e.
137		rg	rg	n. e.
138		right	derecha	DERECHA
139		rl	ll	LL

		U.S.A.	VERSION NUEVA	VERSION TI 99/4A
140		round	redondeo	n. e.
141		rt	de	DA
142		run	cumplir	CUMPLIR
143		save	guardar	GUARDAR
144		savepic	guardardib	n. e.
145		scrunch	proporción	n. e.
146		se	fr	ON
147		sentence	frase	ORACION
148		setbg	fcolorf	CF
149		setc	fcolor	PONERCOLOR
150	#	setcursor	fcursor	PONERMOSAICO
151	*	seth	frumbo	PONERRUMBO
152	#	setpc	fcolorp	PONERCOLOR
153		setpos	fpos	PXY
154		setscr	fproporción	n. e.
155		setsh	ffig	LLEVAR
156		setsp	fvel	PV
157	#	settext	ftexto	PONERMOSAICO
158		setx	fx	PX
159		setxvel	fvelx	PXV
160		sety	fy	PY
161		setyvel	fvely	PYV
162		shade	sombrear	n. e.
163	*	shape	fig	FIGURA
164		show	mostrar	n. e.
165		shownp	visible?	n. e.
166		sin	sen	n. e.
167		snap	convertir	n. e.
168	*	space	espacio	.NODOS
169		speed	vel	n. e.
170		speed	velocidad	VELOCIDAD
171		sqrt	rc	n. e.
172		st	mt	MT
173		stamp	estampar	n. e.
174		stop	parar	PARAR
175		sum	suma	SUMA
176	*	tell	activar	DECIR
177		text	texto	TEXTO
178	*	thaw	descongelar	SEGUIR
179		thing	cosa	COSA
180		to	para	PARA
181	#	toot	sonido	NOTA
182		toplevel	nivelsuperior	n. e.

	U.S.A.	VERSION NUEVA	VERSION TI 99/4A
183	towards	hacia	n. e.
184	true	cierto	CIERTO
185	type	escribirs	ESCRIBIR
186	version	versión	n. e.
187	wait	esperar	ESPERAR
188	when	cuando	CUANDO
189	who	quién	QUIEN
190	window	ventana	n. e.
191	word	palabra	PALABRA
192	wordp	palabra?	n. e.
193	wrap	vuelta	n. e.
194	xcor	coorx	XCOOR
195	xvel	velx	XVEL
196	ycor	coory	YCOOR
197	yvel	vely	YVEL



ANEXO F

EQUIVALENCIAS ENTRE LAS VERSIONES EN ESPAÑOL Y EN INGLÉS

ESPAÑOL-INGLES	ESPAÑOL	INGLES
1	(	(
2	)	)
3	*	*
4	+	+
5	–	–
6	.dentro	.in
7	.depositar	.deposit
8	.examinar	.examine
9	.fuera	.out
10	.llamar	.call
11	/	/
12	<	<
13	=	=
14	>	>
15	activar	tell
16	ad	fd
17	adelante	forward
18	al.contacto	on.touch
19	arctan	arctan
20	ascii	ascii
21	at	bk
22	atrás	back
23	azar	random
24	bg	cg
25	bo	er
26	boarchivo	erf
27	bods	erd
28	bon	ern
29	bons	erns
30	boprops	erprops
31	bops	erps

ESPAÑOL-INGLES	ESPAÑOL	INGLES
32	borrar	erase
33	botodo	erall
34	botón?	buttonp
35	bp	cs
36	bt	ct
37	cada	each
38	cambiar.color	change. colour
39	car	char
40	caracter	char
41	centro	home
42	cierto	true
43	cociente	quotient
44	color	colour
45	color.debajo	colour.over
46	colorf	bg
47	colorp	pc
48	congelar	freeze
49	conimpresora	printer
50	conpluma	pendown
51	convertir	snap
52	coorx	xcor
53	coory	ycor
54	copiardef	copydef
55	copiarfig	copysh
56	cos	cos
57	cosa	thing
58	cp	pd
59	cuando	when
60	cuenta	count
61	cumplir	run
62	cursor	cursor
63	de	rt
64	definido?	definedp
65	definir	define
66	derecha	right
67	descongelar	thaw
68	dif	diff
69	dir	dir
70	distancia	distance
71	ed	ed
72	edfig	es
73	editar	edit
74	edns	edns
75	ent	int
76	esc	pr
77	escribir	print



ESPAÑOL-INGLES	ESPAÑOL	INGLES
78	escribirs	type
79	espacio	space
80	esperar	wait
81	estampar	stamp
82	falso	false
83	fcolor	setc
84	fcolorf	setbg
85	fcolorp	setpc
86	fcursor	setcursor
87	ffig	setsh
88	fig	shape
89	fin	end
90	fpos	setpos
91	fproporción	setscr
92	fr	se
93	frase	sentence
94	frumbo	seth
95	ftexto	settext
96	fvel	setsp
97	fvelx	setxvel
98	fvely	setyvel
99	fx	setx
100	fy	sety
101	guardar	save
102	guardarc	csave
103	guardardib	savepic
104	hacer	make
105	hacia	towards
106	igual?	equalp
107	imarchivo	pofile
108	imds	pods
109	imns	pons
110	imp	po
111	improps	pps
112	imps	pops
113	imtodo	poall
114	imts	pots
115	item	item
116	iz	lt
117	izquierda	left
118	lc	rc
119	limpiar	clean
120	lista	list
121	lista?	listp
122	listaprops	plist

ESPAÑOL-INGLES	ESPAÑOL	INGLES
123	ll	rl
124	mando	joy
125	mando?	joyp
126	menosprimero	butfirst
127	menosúltimo	butlast
128	minus	minus
129	mostrar	show
130	mp	bf
131	mt	st
132	mu	bl
133	nivelsuperior	toplevel
134	no	not
135	nombrar	name
136	nombre?	namep
137	número?	numberp
138	o	or
139	obfig	getsh
140	obprop	gprop
141	ot	ht
142	palabra	word
143	palabra?	wordp
144	para	to
145	parar	stop
146	pb	pe
147	pedir	ask
148	pi	px
149	pintar	fill
150	ponfig	putsh
151	ponpri	fput
152	ponúlt	lput
153	pos	pos
154	pprop	pprop
155	primero	first
156	primitiva?	primitivep
157	primitivas	primitives
158	producto	product
159	proporción	scrunch
160	punto	dot
161	quién	who
162	rc	sqrt
163	reazar	rerandom
164	reciclar	recycle
165	recordar	load
166	recordarc	cload
167	recordardib	loadpic



ESPAÑOL-INGLES	ESPAÑOL	INGLES
168	redondeo	round
169	remprop	remprop
170	repetir	repeat
171	resp	op
172	respuesta	output
173	resto	remainder
174	rg	rg
175	ruido	noise
176	rumbo	heading
177	sen	sin
178	si	if
179	sinimpresora	noprinter
180	sinpluma	penup
181	sombrear	shade
182	sonido	toot
183	sp	pu
184	suma	sum
185	tecla?	keyp
186	texto	text
187	todas	all
188	último	last
189	vacía?	emptyp
190	vel	speed
191	velocidad	speed
192	velx	xvel
193	vely	yvel
194	ventana	window
195	versión	version
196	visible?	shownp
197	vuelta	wrap
198	y	and



INGLES-ESPAÑOL	INGLES	ESPAÑOL
1	(	(
2	)	)
3	*	*
4	+	+
5	–	–
6	.call	.llamar
7	.deposit	.depositar
8	.examine	.examinar
9	.in	.dentro
10	.out	.fuera
11	/	/
12	<	<
13	=	=
14	>	>
15	all	todas
16	and	y
17	arctan	arctan
18	ascii	ascii
19	ask	pedir
20	back	atrás
21	bf	mp
22	bg	colorf
23	bk	at
24	bl	mu
25	butfirst	menosprimero
26	butlast	menosúltimo
27	buttonp	botón?
28	cg	bg
29	change.colour	cambiar.color
30	char	car
31	char	caracter
32	clean	limpiar
33	cload	recordac
34	colour	color
35	colour.over	color.debajo
36	copydef	copiardef
37	copysh	copiarfig
38	cos	cos
39	count	cuenta
40	cs	bp
41	csave	guardarc
42	ct	bt
43	cursor	cursor
44	define	definir
45	definedp	definido?

INGLES-ESPAÑOL	INGLES	ESPAÑOL
46	diff	dif
47	dir	dir
48	distance	distancia
49	dot	punto
50	each	cada
51	ed	ed
52	edit	editar
53	edns	edns
54	emptyp	vacía?
55	end	fin
56	equalp	igual?
57	er	bo
58	erall	botodo
59	erase	borrar
60	erd	bods
61	erf	boarchivo
62	ern	bon
63	erns	bons
64	erprops	boprops
65	erps	bops
66	es	edfig
67	false	falso
68	fd	ad
69	fill	pintar
70	first	primero
71	forward	adelante
72	fput	ponpri
73	freeze	congelar
74	getsh	obfig
75	gprop	obprop
76	heading	rumbo
77	home	centro
78	ht	ot
79	if	si
80	int	ent
81	item	ítem
82	joy	mando
83	joyp	mando?
84	keyp	tecla?
85	last	último
86	left	izquierda
87	list	lista
88	listp	lista?
89	load	recordar
90	loadpic	recordardib

INGLES-ESPAÑOL	INGLES	ESPAÑOL
91	lput	ponúlt
92	lt	iz
93	make	hacer
94	minus	minus
95	name	nombrar
96	namep	nombre?
97	noise	ruido
98	noprinter	sinimpresora
99	not	no
100	numberp	número?
101	on.touch	al.contacto
102	op	resp
103	or	o
104	output	respuesta
105	pc	colorp
106	pd	cp
107	pe	pb
108	pendown	conpluma
109	penup	sinpluma
110	plist	listaprops
111	po	imp
112	poall	imtodo
113	pods	imds
114	pofile	imarchivo
115	pons	imns
116	pops	imps
117	pos	pos
118	pots	imts
119	pprop	pprop
120	pps	improps
121	pr	esc
122	primitivep	primitiva?
123	primitives	primitivas
124	print	escribir
125	printer	conimpresora
126	product	producto
127	pu	sp
128	putsh	ponfig
129	px	pi
130	quotient	cociente
131	random	azar
132	rc	lc
133	recycle	reciclar
134	remainder	resto
135	remprop	remprop



INGLES-ESPAÑOL	INGLES	ESPAÑOL
136	repeat	repetir
137	rerandom	reazar
138	rg	rg
139	right	derecha
140	rl	ll
141	round	redondeo
142	rt	de
143	run	cumplir
144	save	guardar
145	savepic	guardardib
146	scrunch	proporción
147	se	fr
148	sentence	frase
149	setbg	fcolorf
150	setc	fcolor
151	setcursor	fcursor
152	seth	frumbo
153	setpc	fcolorp
154	setpos	fpos
155	setscr	fproporción
156	setsh	ffig
157	setsp	fvel
158	settext	ftexto
159	setx	fx
160	setxvel	fvelx
161	sety	fy
162	setyvel	fvely
163	shade	sombrear
164	shape	fig
165	show	mostrar
166	sownp	visible?
167	sin	sen
168	snap	convertir
169	space	espacio
170	speed	vel
171	speed	velocidad
172	sqrt	rc
173	st	mt



INGLES - ESPAÑOL	INGLES	ESPAÑOL
174	stamp	estampar
175	stop	parar
176	sum	suma
177	tell	activar
178	text	texto
179	thaw	descongelar
180	thing	cosa
181	to	para
182	toot	sonido
183	toplevel	nivelsuperior
184	towards	hacia
185	true	cierto
186	type	escribirs
187	version	versión
188	wait	esperar
189	when	cuándo
190	who	quién
191	window	ventana
192	word	palabra
193	wordp	palabra?
194	wrap	vuelta
195	xcor	coorx
196	xvel	velx
197	ycor	coory
198	yvel	vely



ANEXO G

LISTA DE PRIMITIVAS AGRUPADAS POR FUNCIÓN

Gráficos

1 adelante, ad
2 atrás, at
3 bg
4 bp
5 centro
6 colorf
7 colorp
8 conpluma, cp
9 coorx
10 coory
11 derecha, de
12 distancia
13 fcolorf
14 fcolorp
15 fpos
16 fproporción
17 frumbo
18 fx
19 fy
20 hacia
21 izquierda, iz
22 limpiar
23 mt
24 ot
25 pb
26 pi
27 pos
28 proporción
29 punto
30 rumbo
31 sinpluma, sp
32 ventana
33 visible?
34 vuelta

Palabras y Listas

1 ascii
2 caracter, car
3 cuenta
4 frase, fr
5 igual?
6 item
7 lista
8 lista?
9 menosprimero, mp
10 menosúltimo, mu
11 número?
12 palabra
13 palabra?
14 ponpri
15 ponúlt
16 primero
17 último
18 vacía?

Variables

1 cosa
2 hacer
3 nombrar
4 nombre?



Matemáticas

1 arctan
2 azar
3 cociente
4 cos
5 dif
6 ent
7 minus
8 producto
9 rc
10 reazar
11 redondeo
12 resto
13 sen
14 suma
15 +
16 −
17 *
18 /
19 <
20 =
21 >

Definición de procedimientos

1 copiardef
2 definido?
3 definir
4 editar, ed
5 edns
6 fin
7 para
8 primitiva?
9 texto

Operaciones lógicas

1 no
2 o
3 y

Recordar y guardar

1 guardarc
2 recordarc

Pantalla, teclado e impresora

1 bt
2 conimpresora
3 cursor
4 escribir, esc
5 escribirs
6 esperar
7 fcursor
8 ftexto
9 lc
10 ll
11 mostrar
12 sinimpresora
13 tecla?

Espacio de trabajo

1 bon
2 bons
3 bops
4 borrar, bo
5 botodo
6 espacio
7 imns
8 imp
9 imps
10 imtodo
11 imts
12 primitivas
13 reciclar

Lista de propiedades

1 boprops
2 improps
3 listaprops
4 obprop
5 pprop
6 remprop

Música

1 ruido
2 sonido

Tortugas en movimiento

1 activar
2 al.contacto
3 bods
4 botón?
5 cada
6 cambiar.color
7 color
8 color.debajo
9 congelar
10 convertir
11 copiarfig
12 cuando
13 descongelar
14 edfig
15 estampar
16 fcolor
17 ffig
18 fig
19 fvel
20 fvelx
21 fvely
22 imds
23 mando
24 mando?
25 obfig
26 pedir
27 pintar
28 ponfig
29 quién
30 rg
31 sombrear
32 todas
33 velocidad, vel
34 velx
35 vely

Ordenes de control y condicionales

1 cierto
2 cumplir
3 falso
4 nivelsuperior
5 parar
6 repetir
7 respuesta, resp
8 si

Primitivas del MSX-DOS (sólo para máquinas de 64 K)

1 boarchivo
2 dir
3 guardar
4 guardardib
5 imarchivo
6 recordar
7 recordardib

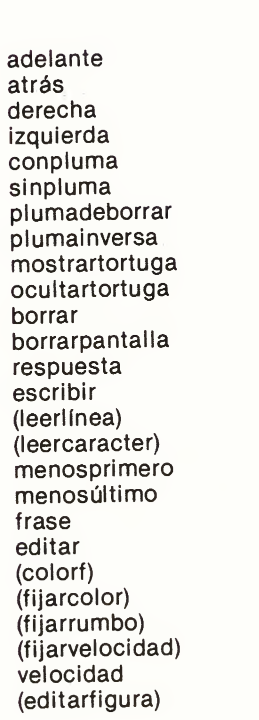
Primitivas especiales

1 .dentro
2 .depositar
3 .examinar
4 .fuera
5 .llamar
6 versión



ANEXO H
ABREVIATURAS

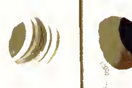




	Castellano	Inglés
adelante	ad	fd
atrás	at	bk
derecha	de	rt
izquierda	iz	lt
conpluma	cp	pd
sinpluma	sp	pu
plumadeborrar	pb	pe
plumainversa	pi	pi
mostrartortuga	mt	st
ocultartortuga	ot	ht
borrar	bo	er
borrarpantalla	bp	cs
respuesta	resp	op
escribir	esc	pr
(leerlínea)	ll	rl
(leercaracter)	lc	rc
menosprimero	mp	bf
menosúltimo	mu	bl
frase	fr	se
editar	ed	ed
(colorf)	colorf	bg
(fijarcolor)	fcolor	setc
(fijarrumbo)	frumbo	seth
(fijarvelocidad)	fvel	setsp
velocidad	vel	speed
(editarfigura)	edfig	es



ÍNDICE





LA CREACION DEL LOGO

El Logo original fue desarrollado por el Dr. Seymour Papert, ganador del prestigioso premio Marconi 1981. Discípulo del epistemólogo suizo Jean Piaget, estudió en Ginebra psicología de la intelingencia y el aprendizaje, en los niños. En el Instituto Tecnológico de Massachusetts (M.I.T) continuó sus investigaciones sobre inteligencia artificial. Allí desarrolló con el grupo Logo nuevos medios tecnológicos de ayuda para la educación. Más adelante el sistema Logo fue aplicado y generalizado en numerosos establecimientos educacionales.
El Dr. Papert, quien ha definido a Logo como un lenguaje para aprender, fundó Logo Computer Systems Inc. en 1980, y con su equipo de investigación desarrolló el lenguaje MSX-LOGO.

APRENDER A APRENDER

Su aprendizaje es sencilo.
Quienes lo han experimentado perciben que la computación no es un misterio.
Al realizar un procedimiento, es frecuente que no funcione bien inicialmente. Se produce entonces una depuración donde se aíslan y corrigen los errores. Este proceso es parte esencial en la comprensión y el aprendizaje de la computación. Las discrepancias entre lo que se intenta hacer y lo que realmente sucede se observan,en este lenguaje, inmediatamente y el error se transforma, así, en fuente de entendimiento.
Este método de trabajo, operable cómodamente con el lenguaje Logo es frecuente en los procesos creativos científicos y artísticos. La forma última evoluciona en la marcha de una idea por asociaciones y deseos preconscientes.
El usuario deja de ser un usuario pasivo, se convierte en el constructor activo de su propio producto, tornando al aprendizaje en un proceso activo con un propósito reconocible.

POR QUE LOGO ES UTIL PARA LA EDUCACION

- Al favorecer la programación estructurada, permite desarrollar las propias estrategias de razonamiento.
- Niños y adultos que aprenden Logo son estimulados por un lenguaje que desafía constantemente su pensamiento.
- Como está escrito en castellano, los niños no deben aprender códigos diferentes ya que se encuentran con su lengua materna
- Logo se encuentra muy cercano al juego, que es la forma privilegiada de expresión en los niños. Ellos pueden inventar sus propios juegos. La creatividad infantil encuentra en este lenguaje un excelente camino para su desarrollo.
- Logo posibilita que padres e hijos lleguen a un entendimiento de las más complejas ideas de la ciencia de la computación y a participar activamente en el desarrollo de esta disciplina.

LENGUAJE LOGO MSX EN CASTELLANO CARACTERISTICAS

- Gráficos de tortugas en 16 colores.
- Edita y almacena hasta 60 formas de tortuga.
- Opera con hasta 30 tortugas simultáneas.
- Detección de colisiones y mandos (Joystick).
- Animación, pintado y sombreado de gráficos.
- Uso optativo de ventana.
- Música y exploración de sonido.
- Matemática de alta precisión.
- Texto y gráficos en toda la pantalla.
- Editor de texto de pantalla completa.
- Procesamiento de palabras y listas.
- Listas de propiedades.
- Almacenamiento en discos o cassettes.



Producido en Argentina por TELEMATICA S.A.